BROCADE

CONFIGURATION GUIDE

Brocade NetIron Switching Configuration Guide

Supporting Multi-Service IronWare R05.9.00b

53-1003837-04
20 May 2016

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names
	Identifies keywords and operands
	Identifies the names of user-manipulated GUI elements
	Identifies text to enter at the GUI
italic text	Identifies emphasis
	Identifies variables
	Identifies document titles
`Courier font`	Identifies CLI output
	Identifies command syntax examples

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional.
	Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
	In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.

Convention	Description
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

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

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

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

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

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

You can download additional publications supporting your product at www.brocade.com. Select the Brocade Products tab to locate your product, then click the Brocade product name or image to open the individual product page. The user manuals are available in the resources module at the bottom of the page under the Documentation category.

To get up-to-the-minute information on Brocade products and resources, go to MyBrocade. You can register at no cost to obtain a user ID and password.

Release notes are available on MyBrocade under Product Downloads.

White papers, online demonstrations, and data sheets are available through the Brocade website.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers contact their OEM/Solutions provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to http://www.brocade.com/services-support/index.html.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • My Cases through MyBrocade • Software downloads and licensing tools • Knowledge Base	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • For areas unable to access toll free number: +1-408-333-6061 • Toll-free numbers are available in many countries.	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/Solution Provider, contact your OEM/Solution Provider for all of your product support needs.

- OEM/Solution Providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/Solution Provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/Solution Provider.

Document feedback

To send feedback and report errors in the documentation you can use the feedback form posted with the document or you can e-mail the documentation team.

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com.
- By sending your feedback to documentation@brocade.com.

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

About This Document

Audience

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

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

Supported hardware and software

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

TABLE 1 Supported devices

Brocade NetIron XMR Series	Brocade MLX Series	NetIron CES 2000 and NetIron CER 2000 Series
Brocade NetIron XMR 4000	Brocade MLX-4	Brocade NetIron CES 2024C
Brocade NetIron XMR 8000	Brocade MLX-8	Brocade NetIron CES 2024F
Brocade NetIron XMR 16000	Brocade MLX-16	Brocade NetIron CES 2048C
Brocade NetIron XMR 32000	Brocade MLX-32	Brocade NetIron CES 2048CX
	Brocade MLXe-4	Brocade NetIron CES 2048F
	Brocade MLXe-8	Brocade NetIron CES 2048FX
	Brocade MLXe-16	Brocade NetIron CER 2024C
	Brocade MLXe-32	Brocade NetIron CER-RT 2024C
		Brocade NetIron CER 2024F
		Brocade NetIron CER-RT 2024F
		Brocade NetIron CER 2048C
		Brocade NetIron CER-RT 2048C
		Brocade NetIron CER 2048CX
		Brocade NetIron CER-RT 2048CX
		Brocade NetIron CER 2048F
		Brocade NetIron CER-RT 2048F
		Brocade NetIron CER 2048FX
		Brocade NetIron CER-RT 2048FX

Supported software

For the complete list of supported features and the summary of enhancements and configuration notes for this release, refer to the *Brocade NetIron Release Notes* .

Notice to the reader

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

These references are made for informational purposes only.

Corporation	Referenced Trademarks and Products
Microsoft Corporation	Internet Explorer
Mozilla Corporation	Mozilla Firefox
Sun Microsystems	Java Runtime Environment

How command information is presented in this guide

Starting with NetIron 5.6.00, command syntax and parameter descriptions are removed from commands that are referenced in configuration tasks. To find the full description of a specific command, including all required and optional keywords and variables, refer to the *NetIron Command Reference* for your software release.

Configuring Interface Parameters

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

NOTE

To modify Layer 2, Layer 3, or Layer 4 features on a port, refer to the appropriate section in this chapter or other chapters.

Assigning a port name

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

To assign a name to a port, enter the following command.

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

Syntax: [no] port-name text

The *text* parameter is an alphanumeric string. The name can have up to 255 characters on a Brocade device and can include blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Assigning an IP address to a port

To assign an IP address to an interface, enter the following commands.

```
device(config)# interface e 1/8
device(config)# ip address 10.45.6.110 255.255.255.0
```

Syntax: [no] ip address ip-addr ip-mask

or

Syntax: [no] ip address ip-addr/mask-bits

> **NOTE**
> You also can enter the IP address and mask in CIDR format, as follows.device `(config)# ip address 10.45.6.1/24`

Modifying port speed

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

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

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

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

Syntax: [no] speed-duplex value

> **NOTE**
> The speed-duplex configuration is applicable to the first four combination ports of the Brocade NetIron CES 2024F-4X module and not applicable to the remaining fiber ports. This is specific to combination ports when the fiber link is connected.

The *value* can be one of the following:

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

The default is auto.

> **NOTE**
> An auto negotiation port must be connected to another auto negotiation port. If you connect an auto negotiation port to a fixed speed or duplex port, the behavior is undefined. Also, ports must be disabled before changing speed.

Modifying port mode

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

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

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

Syntax: speed-duplex value

The *value* can be one of the following:

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

The default is auto.

Auto Negotiation Speed Limit

Auto-negotiation is an active method of determining the link mode. Each interface is expected to transmit specific information in a specific format. If an interface that is expecting to use auto-negotiation does not receive this information from the other side, it assumes the other side cannot detect or change its mode.

One of the most common causes of performance issues on 10/100/1000 Mb Ethernet links occurs when one port on the link operates at half-duplex while the other port operates at full-duplex. This occurs when one or both ports on a link are reset and the auto-negotiation process does not result in both link partners having the same configuration. It also can occur when users reconfigure one side of a link and forgets to reconfigure the other side. Both sides of a link should have auto-negotiation on, or both sides should have it off.

The auto negotiation speed limit feature allows the user to reduce or limit the port speed when auto- negotiation is configured. You can set the port to automatically reduce the speed from 1000Mb to 100Mb or 10Mb. The **down-shift** option will reduce the port speed to 100Mb from 1000Mb automatically once a 2-wire cable is detected.

The auto negotiation speed limit feature is supported only on FIXED (non SFP) copper ports when auto-neg is ON.

```
device(config)# interface e 1/8
device(config-if-e10000-1/8)# auto
device(config-if-e10000-1/8)# link-config gig copper autoneg-control down-shift
```

Syntax: [no] link-config gig copper autoneg-control [down-shift | 100m | 10m]

The *10m* option will limit the port to negotiation to speeds and duplex of 10mb.

The *100m* option will limit the port to negotiation of speeds and duplex below 100mb.

Disabling or re-enabling a port

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

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

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

Syntax: [no] disable

Syntax: [no] enable

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

```
device(config)# interface ve 1
device(config-vif-1)# disable
```

Syntax: [no] disable

To re-enable a virtual routing interface, enter the **enable** command at the Interface configuration level. For example, to re-enable virtual routing interface v1, enter the following command.

```
device(config-vif-1)# enable
```

Syntax: [no] enable

Disabling Source Address Learning on a port

The default operation is for Source Address (SA) Learning to be enabled on all ports. It can be useful to disable SA Learning on a port in situations where high CPU usage is occurring because a large number of packets are being sent to the CPU for SA Learning. For example, it can be useful to disable SA Learning on physical ports that are part of a Virtual Ethernet (VE) interface that has no need to switch packets.

SA Learning can be disabled on a port using the **sa-learning-disable** command as shown in the following.

```
device(config)# interface e 1/8
device(config-if-e10000-1/8)# sa-learning-disable
```

Syntax: [no] sa-learning-disable

Changing the default Gigabit negotiation mode

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

- **neg-full-auto** - The port is only for copper-SFP and to support 10/100/1000M tri-speed auto negotiation.
- **auto-full** -- The port tries to perform a negotiation with its peer port to exchange capability information. If it is unable to reach an agreed upon speed, the port goes into a fixed speed and keeps the link up.
- **auto-gig** - The port tries to performs a negotiation with its peer port to exchange capability information. This is the default state.
- **neg-off** - The port does not try to perform a negotiation with its peer port.

Unless the ports at both ends of a Gigabit Ethernet link use the same mode (either **auto-gig** or **neg-off**), the ports cannot establish a link. An administrator must intervene to manually configure one or both sides of the link to enable the ports to establish the link.

NOTE
Brocade NetIron XMR Series, Brocade NetIron MLX Series, and Brocade MLXe Series support **auto-gig** and **neg-off** options. The **neg-full-auto** and **auto-full** options are not supported on the chassis platforms. Brocade CES and CER-RT series support all four options.

NOTE
Support is provided for the following modules:

- 20x10GE
- 4x10GE-IPSEC

NOTE
Double link flap is observed on the 20x10GE and 4x10GE-IPSEC port modules once the remote peer CER comes back up after reload.

Changing the negotiation mode

You can change the negotiation mode for individual ports as shown in the following.

```
device(config)# interface ethernet 4/1 to 4/4
device(config-mif-4/1-4/4)# gig-default neg-off
```

This command changes the default **auto-gig** setting and sets the negotiation mode to neg-off for ports 4/1 - 4/4.

Use the **auto-gig** command to activate auto-negtotiation.

```
device(config)# interface ethernet 4/1 to 4/4
device(config-mif-4/1-4/4)# gig-default auto-gig
```

Syntax: [no] gig-default neg-full-auto | auto-gig | neg-off | auto -full

Disabling or re-enabling flow control

You can configure full-duplex ports on a system to operate with or without flow control (802.3x).

The command to disable or enable flow control is **flow-control** command. The **flow-control** command is enabled by default.

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

```
Brocade(config)# no flow-control rx-pause-ignore
```

To turn the feature back on, enter the following command.

```
Brocade(config)# flow-control rx-pause-ignore
```

The syntax is given below.

[no] flow-control rx-pause-ignore

Modifying port priority (QoS)

You can give preference to the inbound traffic on specific ports by changing the Quality of Service (QoS) level on those ports. For information and procedures, refer to the Configuring Quality of Service for the Brocade NetIron XMR and Brocade MLX series Chapter in the *Brocade NetIron QoS and Traffic Management Configuration Guide* .

Setting IP VPN packets with a TTL value of 1 to be dropped

This command is for IP VPN packets only. Under normal conditions IP VPN packets with a TTL value equal to 0 are always dropped in hardware regardless of the setting of this command. With this command set, IP VPN packets with TTL value equal to one will also be dropped in hardware.

To enable this command use the following command.

```
device(config)# interface ethernet 4/1 to 4/4
device(config-if-4/1)# hw-drop-bad-ttl-pkt
```

Syntax: [no] hw-drop-bad-ttl-pkt

The default value is off.

Port transition hold timer

Using the **delay-link-event** command will delay the sending of port "up" or "down" events to Layer 2 protocols. While link down events are reported immediately in syslog, their effect on higher level protocols such as OSPF is delayed according to how the delay-link-event is configured. This command affects the physical link events. However, the resulting logical link events are also delayed. This is a per-interface command.

> NOTE
> When a Layer 2 protocol packet is received before the delay-link-event is expired, NetIron will reply to the received Layer 2 protocol without the delay-link-event. After the delay-link-event is expired, NetIron will retransmit the previous Layer 2 event.

For example, if VSRP is enabled on the port, the ownership will not change until the port status has remained up or down for the configured amount of time to ensure that minor transient states of a link do not unintentionally cause a disruptive topology change in the network.

> NOTE
> All LAG ports must have the same delayed-link-down-event configuration.

The following command will delay the sending of port "down" event for 100ms when a port state is detected "down". If the port state is detected "up" afterwards within 100ms, the delayed "down" event is cancelled; otherwise, the "down" event is sent after 100ms. This allows the upper layer applications not to be affected by a port state flapping.

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

Syntax: [no] delay-link-event time up | down

The**time** parameter is the number of 50-ms units. The default is 0. The valid range is from 0 to 200.

The **up** parameter means only "up" events are delayed.

The **down** parameter means that only the down events are delayed.

If neither the **up** or **down** parameter is specified, both up and down events are delayed. This is the default.

Port flap dampening

The port flap dampening feature allows you to configure a wait period before a port, whose link goes down then up, becomes enabled.

If the port link state toggles (from down to up or from up to down) for a specified number of times within a specified period, the interface is physically disabled for the specified wait period. Once the wait period expires, the port's link state is re-enabled. However, if the wait

period is set to zero (0) seconds, or you want to re-enable the port before the wait period expires, the port must be manually re-enabled as described in Re-enabling a port disabled by port link dampening on page 37.

Configuring port link dampening on an interface

This feature is configured at the interface level.

```
device(config)#interface ethernet 2/1
device(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Syntax: [no] link-error-disable toggle-threshold sampling-time-in-sec wait-time-in-sec

The **toggle-threshold** is the number of times a port's link state goes from up to down and down to up before the wait period is activated. The default is 0. Enter a valid value range from 1-50.

The **sampling-time-in-sec** is the amount of time during which the specified toggle threshold can occur before the wait period is activated. The default is 0 seconds. Enter a value between 1 and 65565 seconds.

The **wait-time-in-sec** is the amount of time the port remains disabled (down) before it becomes enabled. Entering 0 indicates that the port will stay down until an administrative override occurs. Enter a value between 0 and 65565 seconds.

Configuring port link dampening on a LAG

You can configure the port link dampening feature on the primary port of a LAG at the interface level using the **link-error-disable** command. Once configured on the primary port of the LAG, the feature is enabled on all port that are members of the LAG. You cannot disable the feature from a member of the LAG.

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

```
device(config)#interface ethernet 2/1
device(config-if-e10000-2/1)#link-error-disable 10 3 10
```

Re-enabling a port disabled by port link dampening

A port disabled by the port link dampening is automatically re-enabled once the wait period expires; however, if the wait period is set to zero (0) seconds or you want to re-enable the port before the configured wait period expires, you must re-enable the port by entering the **link-error-disable** command on the disabled port as shown in the following.

```
device(config)#interface ethernet 2/1
device(config-if-e10000-2/1)#link-error-disable 10 3 10
```

NOTE

You must enter the **link-error-disable** command with the **toggle-thresholdsampling-time-in-sec** and **wait-time-in-sec** variables defined to re-enable the port. Using the **link-error-disable** command without the variables, will not bring the port back up.

Displaying ports configured with port link dampening

Ports that have been disabled due to the port link dampening feature are not identified in a **show running-config** command.

Use the **show interface link-error-disable** to display the ports that have the port link dampening feature enabled.

```
device(config-if-e10000-8/1)#show interfaces link-error-disable
Port 8/1: link-error-disabled (Config: 2 toggles per 3 sec, wait time 1 sec)
Port 8/3: not link-error-disabled (Config: 2 toggles per 2 sec, wait time 30 sec)
Port 8/4: not link-error-disabled (Config: 2 toggles per 2 sec, wait time 30 sec)
```

TABLE 2 link-error-disable

Displays...	Description...
port	The port that has been configured
link-error-disabled	The port that has been disabled by this feature
not link-error-disabled	The "not" means the port has not been disabled due to this feature
toggle	The number of times a port's link state goes from up to down and down to up before the wait period is activated
wait time	The amount of time the port remains disabled (down) before it becomes enabled

Issuing the **disabled-only** with the command displays only the ports that have been disabled by the port link dampening feature.

```
device(config-if-e10000-8/1)#show interfaces link-error-disable disabled-only
Port 8/1: link-error-disabled (Config: 2 toggles per 3 sec, wait time 1 sec)
```

Syntax: show interface link-error-disable [**disabled-only**]

Entering the **show interface link-error-disable** displays all the ports that have the port link dampening feature enabled. Add the **disabled-only** keyword for a list of ports disabled by this feature.

Port loop detection

This feature allows the Brocade device to disable a port that is on the receiving end of a loop by sending test packets. You can configure the time period during which test packets are sent.

Strict mode and Loose mode

There are two types of loop detection; Strict Mode and Loose Mode. In Strict Mode, a port is disabled only if a packet is looped back to that same port. Strict Mode overcomes specific hardware issues where packets are echoed back to the input port. In Strict Mode, loop detection must be configured on the physical port.

In Loose Mode, loop detection is configured on the VLAN of the receiving port. Loose Mode disables the receiving port if packets originate from any port or VLAN on the same device. The VLAN of the receiving port must be configured for loop detection in order to disable the port.

Recovering disabled ports

Once a loop is detected on a port, it is placed in a disabled state. The port will remain disabled until one of the following occurs:

- You manually disable and enable the port at the Interface Level of the CLI
- You enter the command **clear loop-detection** . The **clear loop-detection** command clears the loop detection statistics and enables all disabled ports
- The device automatically re-enables the port. To set your device to automatically re-enable disabled ports, refer to Configuring the device to automatically re-enable ports on page 40.

Disable duration and loop detection interval

By default, the ports are shutdown permanently until user enables it manually. You can configure the disable duration from 1 minute to 1440 minutes (24 hours)

By default. the Loop Detection time Interval between the loop detection BPDU is 1 second. You can configure the loop detection PDU interval from 100ms to 10 seconds.

Configuration notes

Loopback detection packets are sent and received on both tagged and untagged ports. Therefore, this feature cannot be used to detect a loop across separate devices.

The following information applies to Loose Mode loop detection:

- Loop detection is configured on the VLAN. Different VLANs may disable different ports. A disabled port affects every VLAN using it.
- Loose Mode disables the receiving port if packets originate from any port or member port of a VLAN on the same device
- The VLAN of the receiving port must be configured for loop detection in order to disable the port.
- Loose Mode floods test packets to the entire VLAN. This can impact system performance if too many VLANs are configured for Loose Mode loop detection.

The following information applies to Strict Mode loop detection:

- A port is disabled only if a packet is looped back to that same port.
- Loop detection must be configured on the physical port.
- Strict Mode overcomes specific hardware issues where packets are echoed back to the input port.

NOTE

Brocade recommends that you limit the use of Loose Mode. If you have a large number of VLANS or VLAN groups, configuring loop detection on all of them can significantly affect system performance because of the flooding of test packets to all configured VLANs. An alternative to configuring loop detection in a VLAN-group of many VLANs is to configure a separate VLAN with the same tagged port and configuration, and enable loop detection on this VLAN only.

NOTE

When loop detection is used with Layer 2 loop prevention protocols, such as spanning tree (STP), the Layer 2 protocol takes higher priority. Loop detection cannot send or receive probe packets if ports are blocked by Layer 2 protocols, so it does not detect Layer 2 loops when STP is running because loops within a VLAN have been prevented by STP. Loop detection running in Loose Mode can detect and break Layer 3 loops because STP cannot prevent loops across different VLANs. In these instances, the ports are not blocked and loop detection is able to send out probe packets in one VLAN and receive packets in another VLAN. In this way, loop detection running in Loose Mode disables both ingress and egress ports.

Enabling loop detection

Use the loop-detection command to enable loop detection on a physical port (Strict Mode) or a VLAN (Loose Mode). Loop detection is disabled by default. The following example shows a Strict Mode configuration.

```
device(config)#interface ethernet 1/1
device(config-if-e1000-1/1)#loop-detection
```

The following example shows a Loose Mode configuration.

```
device(config)#vlan 20
device(config-vlan-20)#loop-detection
```

The following example shows a Loose Mode configuration for a VLAN group.

```
device(config)#vlan-group 10
device(config-vlan-group-10)#add-vlan 1 to 100
device(config-vlan-group-10)#loop-detection
```

By default, the port will send test packets every one second, or the number of seconds specified by the **loop-detection-interval** command. Refer to Configuring a global loop detection interval on page 40

Syntax: [no] loop-detection

Use the [**no**] form of the command to disable loop detection.

Configuring a global loop detection interval

The loop detection interval specifies how often a test packet is sent on a port. When loop detection is enabled, the loop detection time unit is 0.1 second, with a default of 10 (one second). The range is from 1 (one tenth of a second) to 100 (10 seconds). You can use the show loop-detection status command to view the loop detection interval.

To configure the global loop detection interval, enter a command such as the following.

```
device(config)#loop-detection-interval 50
```

This command sets the loop-detection interval to 5 seconds (50 x 100ms).

To revert to the default global loop detection interval of 10, enter one of the following.

```
device(config)#loop-detection-interval 10
```

OR

```
device(config)#no loop-detection-interval 50
```

Syntax: [no] loop-detection-interval *number*

Where *number*is a value from 1 to 100. The system multiplies your entry by 0.1 to calculate the interval at which test packets will be sent.

Configuring the device to automatically re-enable ports

To configure the Brocade device to automatically re-enable ports that were disabled because of a loop detection, enter the following command The default is 0.

```
device(config)#loop-detection disable-duration 1440
```

The above command will cause the Brocade device to automatically re-enable ports that were disabled for a duration of 24 hours because of a loop detection. This configuration applies to all the ports that are configured the loop detection (strict or loose).

Syntax: [no] loop-detection disable-duration *num*

Use the [**no**] form of the command to disable this feature.

Where *num* is the number of minutes from 0 to 1440. When 0 is specified, it is permanently off.

Clearing loop-detection

To clear loop detection statistics and re-enable all ports that are in disabled state because of a loop detection, enter the following command.

```
device #clear loop-detection
```

Syntax: clear loop-detection [**vlan** | **ethernet**] *vlanid/port-num*

Where **port-num** enables the specified port.

Where *vlan-id* enables all the ports disabled by loop detection for this VLAN

Displaying loop-detection information

Use the **show loop-detection** command to display the loop detection status.

```
device(config-vlan-100)#show loop-detection
loop detection packets interval: 10 (unit 100 msec)
loop detection disable duration: 10 (In minutes, 0 means permanently disabled)
Ports mode loop detection
=========================
port-num   disable-count
1/12       0
1/11       0
Vlan mode loop detection
========================
vlan-id    disable-count
100        2
10          0
200        0
Ports disabled by loop detection
================================
port       age(minutes) disable cause
 1/11  1            Disabled by VLAN: 100 loopdetect 1/11
 1/12  1            Disabled by VLAN: 100 loopdetect 1/12
```

Syntax: show loop-detection

TABLE 3 Port loop detection output description

Parameter	Description
loop detection packets interval	Specifies how often a test packet is sent on a port.
loop detection disable duration	Specifies the device to automatically re-enable ports that were disabled for the configured duration because of a loop detection
ports mode	The VLAN or port that port loop detection was configured on.
loop detection disabled ports	The ports that are disabled by port loop detection. • port - The port number that was disabled by port loop detection. • age - The time duration after which port will be automatically re-enabled. If the age is "0", it means port is not configured to be automatically re-enabled. • disable cause - specifies all the ports that were disabled by loop detection (either strict or loose).

Discarding loop detection frames in the LACP-blocked port

When loop detection is enabled and a loop is detected in the network, the looped packet port is disabled.

In a dynamic LAG scenario on a trunk, loop detection frames are sent out on the active primary port of a trunk group. A packet received in the LACP-blocked port of the transmitting port triggers loop detection on the trunk. Loop detection discards the loop detection frames received in the LACP-blocked port, keeps the port in the up state, and prevents the entire LAG from shutting down.

Syslog message

The following message is logged when a port is disabled due to loop detection. This message will also appear on the console.

```
SYSLOG: Jan 27 18:16:42:<14>Jan 27 18:16:42 LOOP_DETECT LOG: Port Down 1/10 - Loop detected on VLAN: 150
```

Mirroring and Monitoring

You can monitor traffic on Brocade device ports by configuring another port to "mirror" the traffic on the ports you want to monitor. By attaching a protocol analyzer to the mirror port, you can observe the traffic on the monitored ports.

Monitoring traffic on a port is a two-step process:

- Enable a port to act as the mirror port. This is the port to which you connect your protocol analyzer.
- Enable monitoring on the ports you want to monitor.

You can monitor input traffic, output traffic, or both.

Any port on a module can operate as a mirror port and you can configure more than one mirror port. You can configure the mirror ports on different modules and you can configure more than one mirror port on the same module.

Configuration guidelines for monitoring traffic

Use the following considerations when configuring mirroring for inbound and outbound traffic:

- Any port can be mirrored and monitored except for the management port.
- Only one inbound mirror port can be configured for any inbound monitor port.
- Only one outbound mirror port can be configured for any outbound monitor port.
- A LAG port can be configured as either an inbound or outbound monitor port.
- A LAG port cannot be configured as either an inbound or an outbound mirror port.
- Both input and output monitoring are supported.
- Monitoring for LAG ports is supported.
- sFlow and monitoring can be enabled concurrently on the same port.
- ACL-based inbound mirroring is supported.
- ACL-based inbound sFlow is not concurrently supported.
- On the Brocade NetIron CES Series, there can be at most one port configured as the mirror port per port region (aport region is 24-1GbE ports or 2 10-GbE ports). There is no limit on the number of monitor ports that can be configured per port region.

Assigning a mirror port and monitor ports

To configure ethernet port 3/1 for port mirroring, enter the following command.

```
device(config)# mirror-port ethernet 3/1
```

Syntax: [no] mirror-port ethernet slot/portnum

> NOTE
> If a port is configured as a mirror port, all traffic sent from that port will retain the encapsulation of the port being monitored and not add the encapsulation of the Egress port.

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

```
device(config)# interface ethernet 4/1
device(config-if-4/1)# monitor ethernet 3/1
```

Syntax: [no] monitor ethernet slot/portnum both | input | output

Enter the slot and port number of the port that will serve as the monitor port. This port cannot be the same as the mirror port.

NOTE
A mirror port must be an Ethernet port.

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

NOTE
In VPLS, when an unknown unicast traffic is handled, it uses the corresponding VLAN Forwarding ID to flood the packets to the VLAN domain which contains both the monitored port as well as the mirroring port. But in VLL, there is no such flood handling mechanism and hence, there is a discrepancy in the output of the **show statistic brief** command in terms of the **Packet Transmit** count on the mirroring port.

Displaying mirror and monitor port configuration

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

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

Syntax: show monitor config

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

```
device# show monitor actual
Monitored Port 3/1
    Output traffic mirrored to: 1/1
Monitored Port 4/1
    Input traffic mirrored to: 1/2
```

Syntax: show monitor actual

This output displays the output traffic mirrored to mirror port 1/1 from port 3/1 and input traffic mirrored to mirror port 1/2 from port 4/1, which are explicitly configured.

ACL-based inbound mirroring

The Multi-Service IronWare software supports using an ACL to select traffic for mirroring from one port to another. Using this feature, you can monitor traffic in the mirrored port by attaching a protocol analyzer to it.

Considerations when configuring ACL-based inbound mirroring

The following must be considered when configuring ACL-based inbound mirroring:

- Configuring a common destination ACL mirror port for all ports of a PPCR (see below)
- Support with ACL CAM sharing enabled (see below)
- The **mirror** and **copy-sflow** keywords are mutually exclusive on a per-ACL clause basis.
- ACL-based inbound mirroring and port-based inbound mirroring are mutually exclusive on a per-port basis.
- ACL-based mirroring must be configured at the LAG level for individual LAG member ports.
- Configuring ACL-based mirroring at the port level on the primary port of a LAG mirrors all traffic on that LAG to the monitor port.

Configuring a Common Destination ACL mirror port for all ports of a PPCR

All ports using the same PPCR must have a common destination ACL mirror port when configuring ACL-based inbound mirroring. For Example, where ports 4/1 and 4/2 belong to the same PPCR, the following configuration that configures them with different destination ACL mirror ports will fail and generate an error message as shown.

```
device(config)# interface ethernet 4/1
device(config-if-e10000-4/1)# acl-mirror-port ethernet 6/1
device(config-if-e10000-4/1)# interface ethernet 4/2
device(config-if-e10000-4/2)# acl-mirror-port ethernet 6/2
Error: 4/2 and 4/1 should have the same ACL mirror port
```

Support with ACL CAM sharing enabled

For ACL CAM sharing to function, either one of the following conditions must be true:

- All ports that belong to a PPCR have the **acl-mirror-port** command configured to direct mirrored traffic to the same port.
- None of the ports that belong to the PPCR have the **acl-mirror-port** command configured.

ACL CAM sharing cannot function with the configuration shown in the following example because port 4/1 has ACL port mirroring configured and port 4/2 does not.

```
device(config)# enable-acl-cam-sharing
device(config)# interface ethernet 4/1
device(config-if-e10000-4/1)# ip access-group 101 in
device(config-if-e10000-4/1)# acl-mirror-port ethernet 6/1
device(config-if-e10000-4/1)# interface ethernet 4/2
device(config-if-e10000-4/2)# ip access-group 101 in
```

Configuring ACL-based inbound mirroring

The following sections describe how to configure ACL-based Inbound Mirroring on a Brocade device:

- Creating an ACL with a mirroring clause
- Applying the ACL to an interface
- Specifying a destination mirror port
- Specifying the destination mirror port for physical ports
- Specifying the destination mirror port for a LAG
- Configuring ACL-based mirroring for ACLs bound to virtual interfaces
- Specifying the destination mirror port for IP receive ACLs

Creating an ACL with a mirroring clause

The **mirror** keyword in IPv4, Layer 2 and IPv6 ACL clauses directs traffic that matches the clause criteria to be mirrored to another port. In the following examples, the ACL is used to direct IP traffic to a mirror port.

: ACL-based Mirroring Supported for IPv4 ACLs.

```
device(config)# access-list 101 permit ip any any mirror
device(config)# access-list 101 permit ip any any
```

: ACL-based Mirroring supported for IPv6 Inbound ACLs.

```
device(config)# ipv6 access-list gem
device(config-ipv6-access-list gem)# permit tcp 2001:DB8::/64 2001:DB8::/64 mirror
device(config-ipv6-access-list gem)# permit udp 1000:1::/64 2000:1::/64 mirror
device(config-ipv6-access-list gem)# permit icmp 1000:1::/64 2000:1::/64 mirror
device(config-ipv6-access-list gem)# permit ipv6 any any
```

: ACL-based Mirroring supported for Layer-2 Inbound ACLs.

```
device(config)# access-list 400 permit 0000.0000.0010
ffff.ffff.ffff 0000.0000.0020 ffff.ffff.ffff any mirror
device(config)# access-list 400 permit 0000.0000.0050
ffff.ffff.ffff 0000.0000.0020 ffff.ffff.ffff any mirror
device(config)#access-list 400 permit any any any
```

The **mirror** parameter directs selected traffic to the mirrored port. Traffic can only be selected using the **permit** clause. The mirror parameter is supported on rACLs.

> NOTE
> As with any ACL, the final clause must permit desired traffic to flow: be sure to add an appropriate **permit any any** clause to the end of any ACL intended to mirror (and not filter) traffic. Failure to include the **permit** clause will result in disruption of traffic through any interface to which the ACL is applied.

Applying the ACL to an interface

You must apply the ACL to an interface using the **ip access-group command** as shown in the following.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# ip access-group 101 in
```

Specifying the destination mirror port

You can specify physical ports or a LAG to mirror traffic from. The following sections describe how to perform each of these configurations.

Specifying the destination mirror port for physical ports

You must specify a destination port for traffic that has been selected by ACL-based Inbound Mirroring. This configuration is performed at the Interface Configuration of the port whose traffic you are mirroring. In the following example, ACL mirroring traffic from port 1/1 is mirrored to port 1/3.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# acl-mirror-port ethernet 1/3
```

You can also use the ACL-mirroring feature to mirror traffic from multiple ports to a single port using the Multiple Interface Configuration (MIF) mode as shown in the following example.

```
device(config)# interface ethernet 1/1 to 1/2
device(config-mif-e10000-1/1-1/2)# acl-mirror-port ethernet 1/3
```

Syntax: [no] acl-mirror-port ethernet [**slot/port**]

The [slot/port] variable specifies port that ACL-mirror traffic from the configured interface will be mirrored to.

Specifying the destination mirror port for a LAG

You can mirror the traffic that has been selected by ACL-based inbound mirroring from all ports in a LAG by configuring a destination (monitor) port for the LAG at the interface configuration level of the LAG's primary port. Configuring mirroring on the primary port of the LAG causes ACL-selected traffic from all ports in the LAG (including any ports subsequently added to the LAG dynamically on the Brocade NetIron XMR Series and Brocade NetIron MLX Series) to be mirrored to the monitor port. For example, in the following configuration all traffic on LAG "mylag" will be mirrored to port 10/4:

```
device(config)# lag mylag static
device(config-lag-mylag)# ports ethernet 10/1 to 10/3
device(config-lag-mylag)# primary-port 10/1
device(config-lag-mylag)# deploy
```

```
device(config-lag-mylag)# exit
device(config)# interface ethernet 10/1
device(config-if-e1000-10/1)# acl-mirror-port ethernet 10/4
```

Syntax: [no] acl-mirror-port ethernet slot/port

The **ethernet***slot/port* variable specifies the port that ACL-mirror traffic from the LAG will be mirrored to.

The following considerations apply when configuring ACL-based mirroring with LAGs:

- You must configure ACL-mirroring for an individual member port from the LAG configuration level. Attempting to configure ACL-mirroring at the interface level for an individual member port will fail and display the following message.

```
Error: please use  config level to configure ACL based mirroring on  port.
```

- If an individual port is configured for ACL-based mirroring, you cannot add it to a LAG. If you want to add it to a LAG, you must remove it from ACL-based mirroring first. Then you can add it to a LAG. It can then be configured for either ACL-based LAG mirroring or for mirroring an individual port within a LAG.

If you attempt to add a port that is configured for ACL-based mirroring to a LAG, the following message will display.

```
ACL port is configured on port 2/1, please remove it and try again.
 transaction failed:  Config Vetoed
```

- When a LAG with ACL-based mirroring configured on it is deleted or not deployed, the ACL-based mirroring configuration is removed from each of the individual ports that made up the LAG, including the primary port.

Configuring ACL-based mirroring for ACLs bound tovirtual interfaces

For configurations that have an ACL bound to a virtual interface, you must configure the **acl-mirror-port** command on a port for each PPCR that is a member of the virtual interface. For example, in the following configuration ports 4/1 and 4/2 share the same PPCR while port 4/3 uses another PPCR.

```
device(config)# vlan 10
device(config-vlan-10)# tagged ethernet 4/1 to 4/3
device(config-vlan-10)# router-interface ve 10
device(config)# interface ethernet 4/1
device(config-if-e10000-4/1)# acl-mirror-port ethernet 5/1
device(config)# interface ve 10
device(config-vif-10)# ip address 10.10.10.254/24
device(config-vif-10)# ip access-group 102 in
device(config)# access-list 101 permit ip any any mirror
```

In this configuration, the **acl-mirror-port** command is configured on port 4/1 which is a member of ve 10. Because of this, ACL-based mirroring will apply to VLAN 10 traffic that arrives on ports 4/1 and 4/2. It will not apply to VLAN 10 traffic that arrives on port 4/3 because that port uses a different PPCR than ports 4/1 and 4/2. To make the configuration apply ACL-based mirroring to VLAN 10 traffic arriving on port 4/3, you must add the following command to the configuration.

```
device(config)# interface ethernet 4/3
device(config-if-e10000-4/3)# acl-mirror-port ethernet 5/1
```

If the ve contains LAG ports, configuration of acl-mirror-port command on an individual LAG port will also apply to other LAG ports that are in the same PPCR. For example, in the following configuration the acl-mirror-port command is configured for LAG port 10/2, which is a member of ve.

```
device(config)# lag mylag static
device(config-lag-mylag)# ports ethernet 10/1 to 10/4
device(config-lag-mylag)# primary-port 10/1
device(config-lag-mylag)# deploy
device(config-lag-mylag)# acl-mirror-port ethe-port-monitored 10/2 ethe 11/3
device(config)# vlan 10
device(config-vlan-10)# tagged ethe 10/1 to 10/4
device(config-vlan-10)# router-interface ve 10
```

The ACL-based mirroring will apply to VLAN 10 traffic incoming on ports 10/1 and 10/2 since they are in the same PPCR and are members of a virtual interface. However it will not apply to VLAN 10 traffic incoming on 10/3 and 10/4 since they are in a different PPCR. To apply ACL-based mirroring on VLAN 10 traffic incoming on 10/3 and 10/4, you will have to additionally configure the acl-mirror-port ether-port-monitored 10/3 ethe 11/3 command under the LAG.

Specifying the destination mirror port for IP Receive ACLs

When specifying a destination port for IP Receive ACLs, you must configure the **acl-mirror-port** command on all ports supported by the same PPCR. For example, if you are using mirroring traffic for an rACL on a 4 x 10G interface module and you want to mirror traffic incoming on the first PPCR, you have to configure the **acl-mirror-port** command on both ports 1 and 2. If you want to mirror IP Receive ACL permit traffic incoming on all ports of the module, you have to configure the **acl-mirror-port** command on all ports of the module.

10G WAN PHY fault and performance management

This feature provides fault and performance management features such as alarm detection, alarm generation, and performance monitoring on 10 GbE WAN PHY interfaces. It only applies to 10 GbE interfaces configured in the WAN PHY mode.

Using this feature, you can gather fault and performance management information and display it for the current 15 minute interval or for any of the previous 15 minute intervals. In addition, this feature allows you to create a path trace between WAN PHY interfaces to ensure correct connection.

- 8x10G WAN PHY has been supported in releases before 5.8.00b.
- 20x10G WAN PHY support is available starting with 5.8.00b.

Setting a 10 GbE interface to WAN PHY mode

To set a 10 GbE interface to WAN PHY mode, use the following command.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# phy-mode wan
```

Syntax: [no] phy-mode [**wan** | **28k**]

The **wan** parameter sets the PHY mode to WAN.

The **28k** parameter sets the PHY mode to 28k (to allow interoperability with other devices).

The default setting is LAN PHY mode; to reset PHY mode to LAN, use the command **no phy-mode** .

Turning alarm interfaces on and off

When a 10 GbE port is to WAN PHY mode, alarm monitoring is set on by default. You can turn alarm monitoring off for an individual port as shown in the following.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# no alarm-monitoring
```

Syntax: [no] alarm-monitoring

Configuring path trace

You can configure a character string to be carried in the SONET overhead as a method of detecting mis-connection of ports between two devices connected over the WAN PHY. The devices compare the configured character string with the received character string to determine if the connection is valid.

You can configure a Brocade device with the character string "test1" using the following command.

```
device(config)# interface ethernet 3/1
device(config-if-e10000-3/1)# overhead j0-transmit test1
```

Syntax: [no] overhead [**j0-transmit string**] | [**j1-transmit string**]

The *string* variable is the character string used to detect mis-connection of ports it must be configured with the same value on each side of the WAN PHY connection.

Displaying status of alarms on an interface

You can display the current status of WAN PHY alarms as shown in the following.

```
device# show controller curr15min e 3/1
----------------------------------------------------------
10g wan phy alarms statistics  - PORT e3/1
----------------------------------------------------------
ACTIVE ALAMRS : LOS
ACTIVE DEFECTS : LOF-S AIS-L AIS-P
Elapsed time [0 min 13 secs]
Format [alarm type = count]
FM-PARAMS
 Section
  LOS = 1  LOF = 1
Line
  AIS-L = 1 RDI-L = 0
Path
  AIS-P = 1  LOP = 0 PLM = 0 AIS-PFE = 0 PLM-PFE = 0
PM-PARAMS
 Section
  CV = 2  ES = 1  SES = 0  SEFS = 0
 Line
  CV = 3  ES = 1  SES = 0 UAS = 0
  CV-FE  = 0  ES-FE = 0  SES-FE = 0 UAS-FE = 0
 Path
   CV = 4  ES = 1  SES = 0 UAS = 0
   CV-FE  = 0  ES-FE = 0  SES-FE = 0 UAS-FE = 0
```

Syntax: show controller [**curr15min port no** | **slot no**] | [**day port no** | **slot no**] | [**prev15min interval port no** | **slot no**]

The **curr15min** parameter specifies that you want to display WAN PHY alarm and performance information for the current 15 minute interval for either the port number *portno* or slot number *slot no* specified.

The **day** parameter specifies that you want to display WAN PHY alarm and performance information for the current day for either the port number *port no* or slot number *slot no* specified.

The **prev15min** parameter specifies that you want to display WAN PHY alarm and performance information for the a past 15 minute interval as specified by the *interval* variable for either the port number *port no* or slot number *slot no* specified. Possible values for *interval* are 1 - 31 and indicates which previous 15 minute interval you want to display information from. The closest previous interval is 1 and the farthest is 31.

The *port no* variable specifies the port that you want to display WAN PHY alarm and performance information for.

The *slot no* variable specifies the slot that you want to display WAN PHY alarm and performance information for.

This display shows the following information.

TABLE 4 WAN PHY display parameters

Parameter	Description.
Loss of signal (LOS)	LOS is raised when the synchronous signal (STS-N) level drops below the threshold at which a BER of 1 in 103 is predicted. It could be due to a cut cable, excessive attenuation of the signal, or equipment fault. LOS state

TABLE 4 WAN PHY display parameters (continued)

Parameter	Description.
	clears when two consecutive framing patterns are received and no new LOS condition is detected.
Out of frame (OOF) alignment or SEF (Severely errored Frame)	OOF state occurs when four or five consecutive SONET frames are received with invalid (errored) framing patterns (A1 and A2 bytes). The maximum time to detect OOF is 625 microseconds. OOF state clears when two consecutive SONET frames are received with valid framing patterns.
Loss of frame (LOF) alignment	LOF state occurs when the OOF state exists for a specified time in milliseconds. LOF state clears when an in-frame condition exists continuously for a specified time in milliseconds.
Loss of pointer (LOP)	LOP state occurs when N consecutive invalid pointers are received or N consecutive new data flags (NDFs) are received (other than in a concatenation indicator), where N = 8, 9, or 10. LOP state clears when three equal valid pointers or three consecutive AIS indications are received. LOP can be identified as follows: • STS path loss of pointer (SP-LOP) • VT path loss of pointer (VP-LOP)
Alarm indication signal (AIS)	The AIS is an all-ones characteristic or adapted information signal. It is generated to replace the normal traffic signal when it contains a defect condition in order to prevent consequential downstream failures being declared or alarms being raised. Line AIS defect is detected as a "111" pattern in bits 6, 7, and 8 of the K2 byte in five consecutive frames. Line AIS defect is terminated when bits 6, 7, and 8 of the K2 byte do not contain the code "111" for five consecutive frames. STS-Path AIS defect is detected as all ones in bytes H1 and H2 in three contiguous frames. STS-Path AIS defect is terminated when a valid STS Pointer is detected with the NDF set to "1001" (inverted) for one frame, or "0110" (normal) for three contiguous frames. AIS can also be identified as follows: • Line alarm indication signal (AIS-L) • STS path alarm indication signal (SP-AIS) • VT path alarm indication signal (VP-AIS)
Remote error indication (REI)	This is an indication returned to a transmitting node (source) that an errored block has been detected at the receiving node (sink). This indication was formerly known as far end block error (FEBE). REI can also be identified as the following: • Line remote error indication (REI-L) • STS path remote error indication (REI-P) • VT path remote error indication (REI-V)
Remote defect indication (RDI)	This is a signal returned to the transmitting terminating equipment upon detecting a loss of signal, loss of frame, or AIS defect. RDI was previously known as FERF. RDI can also be identified as the following: • Line remote defect indication (RDI-L) • STS path remote defect indication (RDI-P) • VT path remote defect indication (RDI-V)

TABLE 4 WAN PHY display parameters (continued)

Parameter	Description.
B1 error (coding violation, CV)	Parity errors evaluated by byte B1 (BIP-8) of an STS-N are monitored. If any of the eight parity checks fail, the corresponding block is assumed to be in error.
B2 error (coding violation, CV)	Parity errors evaluated by byte B2 (BIP-24 x N) of an STS-N are monitored. If any of the N x 24 parity checks fail, the corresponding block is assumed to be in error.
B3 error (coding violation, CV)	Parity errors evaluated by byte B3 (BIP-8) of a VT-N (N = 3, 4) are monitored. If any of the eight parity checks fail, the corresponding block is assumed to be in error.
Errored Seconds (ES)	At each layer, an Errored Second (ES) is a second with one or more Coding Violations at that layer OR one or more incoming defects (e.g., SEF, LOS, AIS, LOP) at that layer has occurred. Far end - This is an indication returned to a transmitting node (source) that an errored block has been detected at the receiving node (sink). And Errored seconds - far end indicate this error in terms of errored seconds. ES can be identified as follows: • Section Errored seconds (ES-S)· • Line Errored seconds (ES-L), Line Errored seconds- Far end (ES-LFE)· • Path Errored seconds (ES-P), Path Errored seconds- Far end (ES-PFE)
Severely Errored seconds (SES)	At each layer, an Severely Errored Second (SES) is a second with x or more CVs at that layer, or a second during which at least one or more incoming defects at that layer has occurred. Values of x vary depending on the line rate and the Bit Error Rate. SES can be identified as follows: • Section Severely Errored seconds (SES-S)· • Line Severely Errored seconds (SES-L), Line Errored seconds-Far end (SES-LFE)· • Path Severely Errored seconds (SES-P), Path Errored seconds-Far end (SES-PFE)
Severely errored frame seconds (SEFS)	A Severely Errored Framing Second (SEFS) is a seconds with containing one or more SEF events. This counter is only counted at the Section Layer.
Unavailable seconds (UAS)	At the Line, Path, and VT layers, an unavailable second is calculated by counting the number of seconds that the interface is unavailable. At each layer, the SONET or SDH interface is said to be unavailable at the onset of 10 contiguous SESs. The 10 SESs are included in unavailable time. Once unavailable, the SONET or SDH interface becomes available at the onset of 10 contiguous seconds with no SESs. The 10 seconds with no SESs are excluded from unavailable time. With respect to the SONET or SDH error counts at each layer, all counters at that layer are incriminated while the SONET or SDH interface is deemed available at that layer. While the interface is deemed unavailable at that layer, the only count that is incriminated is UASs at that layer. UAS can be identified as follows: • Line Unavailable seconds (UAS-L), Line Unavailable seconds at far end (UAS-LFE)· • Path Unavailable seconds (UAS-P), Path Unavailable seconds (UAS-PFE)

Wait for all cards feature

During a system reload, an Interface module comes up after it completes its initialization process. After an Interface module is up, its ports can come up. Since 10G modules have more packet processors to initialize, 1G ports are up earlier than 10G ports.

> NOTE
> Rebooting interface modules manually is not supported. The wait for all cards feature will only take effect when the entire router or switch is rebooted.

The **wait-for-all-cards** command directs all ports to come up at the same time. This is done by waiting for all Interface modules to come up first, before allowing for ports to come up. This command is shown in the following.

```
device(config)# wait-for-all-cards
```

Syntax: [no] wait-for-all-cards

> NOTE
> With the **wait-for-all-cards** command enabled,10G ports will come up before 1G ports because Multi-Service IronWare software processes 10G port's state changes first.

Link fault signaling

You can enable link fault signaling on 10 or 100 gigabit interfaces. Link fault signaling (LFS) is a physical layer protocol that enables communication on a link between two 10 or 100 Gigabit Ethernet devices. When configured on a Brocade 10 or 100 Gigabit Ethernet port, the port can detect and report fault conditions on transmit and receive ports.

If LFS is configured on an interface, the following Syslog messages are generated when that interface goes up or down or when the TX or RX fiber is removed from one or both sides of the link that has LFS configured:

- SYSTEM: port 2/1 is down (remote fault)
- SYSTEM: Interface ethernet 2/1, state down - remote fault
- SYSTEM: Interface ethernet 2/1, state up

Traditionally, in Brocade NetIron MLX Series and Brocade NetIron XMR Series devices, LFS was disabled in both TX and RX directions. The **link-fault-signaling** command was used to enable LFS in both TX and RX directions. When RX LFS is enabled, a port will be brought up only when the PHY-MAC link is up, and there is no link fault received by the MAC. When RX LFS is disabled, a port will be brought up as long as the PHY-MAC link is up, regardless of any RX fault indication to MAC.

The RX LFS is always enabled by default and cannot be disabled. The **link-fault-signaling** command only applies to enabling or disabling the TX LFS. While RX LFS is recommended to be enabled at all times, for some applications it is requested to have the means to disable RX LFS.

There are two independent link-fault signaling commands **link-fault-signaling** and **link-fault-signaling ignore-rx.** These commands are applicable at both the global (system-level) and per-port level. Both global and per-port configurations are considered jointly to determine the resulting per-port configuration. When a global configuration is applied, it will override the corresponding per-port configuration already present. It is recommended to configure the global configuration prior to applying per-port configurations.

To configure LFS, enter the following commands.

```
device(config)# interface ethernet 1/4
device(config-if-e1000-1/4)# link-fault-signaling
```

Syntax: [no] link-fault-signaling

LFS is disabled by default.

NOTE
Ensure both sides are LFS ON when using LFS with RX (always on) and another router (which can be configured ON or OFF). Do not assume all boxes have LFS ON or OFF by default. Be sure and check.

To to disable RX LFS on a specified port, enter the **link-fault-signaling ignore-rx** command.

```
device(config)# interface ethernet 1/4
device(config-if-e1000-1/4)# link-fault-signaling ignore-rx
```

Syntax: [no] link-fault-signaling ignore-rx

RX LFS is ignored on the specified port.

Configuration Examples

The following configuration examples show global and port configurations.

```
device(config)# link-fault-signaling
Brocade(config)#show run
Current configuration:
!
ver V5.4.0iT163
module 1 ni-mlx-8-port-10g-m
module 3 ni-mlx-8-port-10g-m
!
link-fault-signaling
!3  3ffff(R) 0.0.0.0/0          N/A            Dis N/A  Drop   00094
```

TX LFS and RX LFS are enabled on all ports.

```
device(config)# interface e 3/1
Brocade(config-if-e100000-3/1)#link-fault-signaling ignore-rx
Brocade(config-if-e100000-3/1)#show run
Current configuration:
!
ver V5.4.0iT163
module 1 ni-mlx-8-port-10g-m
module 3 ni-mlx-8-port-10g-m
!
link-fault-signaling
!
interface ethernet 3/1
link-fault-signaling ignore-rx
!
```

TX LFS is enabled on all ports. RX LFS is enabled on all ports except 3/1

Port configuration overwritten by global configuration

```
device(config)#show run
Current configuration:
!
ver V5.3.0pT183
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
!
end
device(config)#show link-fault-signaling
Global Link Fault : RX ON   TX OFF
PORT   #:  LINK FAULT:
PORT 2/1:  RX ON   TX OFF
PORT 2/2:  RX ON   TX OFF
```

TX LFS is disabled on all ports and RX LFS is enabled on all ports.

```
device(config)#link-fault-signaling
device(config)#show run
Current configuration:
!
ver V5.3.0pT183
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
link-fault-signaling
!
end
device(config)#show link-fault-signaling
Global Link Fault : RX ON   TX ON
PORT   #:  LINK FAULT:
PORT 2/1:  RX ON   TX ON
PORT 2/2:  RX ON   TX ON
```

TX LFS is enabled on all ports and RX LFS is enabled on all ports.

```
device(config)#int e 2/1
device(config-if-e10000-2/1)#no link-fault-signaling
device(config-if-e10000-2/1)#show run
Current configuration:
!
ver V5.3.0pT183
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
link-fault-signaling
!
interface ethernet 2/1
 no link-fault-signaling
!
end
device(config-if-e10000-2/1)#show link-fault-signaling
Global Link Fault : RX ON   TX ON
PORT   #:  LINK FAULT:
PORT 2/1:  RX ON   TX OFF
PORT 2/2:  RX ON   TX ON
```

TX LFS is enabled on all ports except 2/1 and RX LFS is enabled on all ports.

```
device(config-if-e10000-2/1)#exit
device(config)#link-fault-signaling
device(config)#show run
Current configuration:
!
ver V5.3.0pT183
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
link-fault-signaling
!
end
device(config)#show link-fault-signaling
Global Link Fault : RX ON   TX ON
PORT   #:  LINK FAULT:
```

```
PORT 2/1:  RX ON   TX ON
PORT 2/2:  RX ON   TX ON
```

TX LFS is enabled on all ports and RX LFS is enabled on all ports. The previously configured no link-fault-signaling on port 2/1 is overwritten by the global TX LFS enable.

Configuring RX LFS on all ports and enabling TX LFS on one port

```
device(config)#show run
Current configuration:
!
ver V5.3.0pT183
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
!
end

device(config)#show link-fault-signaling
Global Link Fault : RX ON   TX OFF
PORT   #:  LINK FAULT:

PORT 2/1:  RX ON   TX OFF
PORT 2/2:  RX ON   TX OFF
```

TX LFS is disabled on all ports and RX LFS is enabled on all ports.

```
device(config)#int e 2/1
device(config-if-e10000-2/1)#link-fault-signaling
device(config-if-e10000-2/1)#exit
device(config)#show run
Current configuration:
!
ver V5.3.0pT183
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
!
interface ethernet 2/1
 link-fault-signaling
!
end

device(config)#show link-fault-signaling
Global Link Fault : RX ON   TX OFF
PORT   #:  LINK FAULT:

PORT 2/1:  RX ON   TX ON
PORT 2/2:  RX ON   TX OFF
```

TX LFS is enabled only on port 2/1 and RX LFS is enabled on all ports.

Configuring TX LFS on all ports and enabling RX LFS on all ports except one port

```
device(config)#link-fault-signaling
device(config)#no link-fault-signaling ignore-rx
device(config)#interface e 2/1
device(config-if-e10000-2/1)#link-fault-signaling ignore-rx
device(config-if-e10000-2/1)#exit
device(config)#show run
Current configuration:
!
ver V5.3.0pT183
!
```

```
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
hostname Brocade
link-fault-signaling
!
interface ethernet 2/1
 link-fault-signaling ignore-rx
!
end

device(config)#show link-fault-signaling
Global Link Fault : RX ON   TX ON
PORT   #:  LINK FAULT:

PORT 2/1:  RX OFF  TX ON
PORT 2/2:  RX ON   TX ON
```

TX LFS is enabled on all ports and RX LFS is enabled on all ports except 2/1.

Displaying link-fault-signaling information

You can display information for link-fault-signaling in a Brocade device by using the **show link-fault-signaling** command.

To display if LFS is configured on an interface, enter the following command.

```
device# show link-fault-signaling
Global Link Fault : RX ON   TX OFF
PORT   #:  LINK FAULT:
PORT 2/1:  RX ON   TX OFF
PORT 2/2:  RX ON   TX OFF
PORT 2/3:  RX ON   TX OFF
PORT 2/4:  RX ON   TX OFF
PORT 2/5:  RX ON   TX OFF
PORT 2/6:  RX ON   TX OFF
PORT 2/7:  RX ON   TX OFF
PORT 2/8:  RX ON   TX OFF
PORT 3/1:  RX ON   TX OFF
PORT 3/2:  RX ON   TX OFF
PORT 3/3:  RX ON   TX OFF
PORT 3/4:  RX ON   TX OFF
PORT 3/5:  RX ON   TX OFF
PORT 3/6:  RX ON   TX OFF
PORT 3/7:  RX ON   TX OFF
PORT 3/8:  RX ON   TX OFF
```

NOTE

The **show link-fault-signaling** command does not display RX and TX information for 1 Gb Ethernet ports.

Displaying and clearing remote fault counters

To display Remote Fault Notification (RFN) counters on 10GbE LAN physical interface, enter the following command.

device # show remote-fault ethernet 1/1 to 1/4

```
Port RFN Detected Remote-fault count time last RFN detected

---- ----------------- ---------------------- --------------------------

1/1 Yes 15 Sep 29 22:03:03

1/2 No 0 -

1/3 No 12 Aug 20 13:22:14
```

```
1/4 No 0 -

    ** remote-fault counters are only supported for ports in LAN PHY mode on 10GE modules. **
```

The example above displays remote fault notification counters with slot 1 as a 10GbE module, and ports 1/1, 1/2, 1/3, and 1/4 in LAN mode.

If the user enters a slot number that is not a 10 GbE port, or if any port in the port range is not a 10GbE port in LAN mode, the following error message is displayed.

```
device# show remote-fault slot 3
remote-fault counters are only supported for ports in LAN PHY mode on 10 GE modules.
```

To clear remote fault notification counters on a 10 GbE LAN physical interface, enter the following command.

```
device#clear remote-fault slot 1
```

Syntax: show or clear remote-fault [**ethernet slot#/port#** [**to slot#/port**] | **slot slot#**]

You can display information for remote fault notification counters in a Brocade device by using the **show remote-fault** command without options,

Use the ethernet *<slot#/port#>* option to limit the display to a single ethernet port.

Use the to *<slot#/port>* option for a range of ports.

Use the slot *<slot#>* option to limit the display to a single slot.

The following table describes the output of the **show remote-fault** command

TABLE 5 Display of show remote-fault output

This field...	Displays...
Port	The *<port#>* variable specifies the port number for the interface module.
RFN Detected	The remote-fault notification is detected on a given interface. If "Yes" is displayed, then the remote-fault notification is detected on the given port at the time of inquiry. If "No" is displayed, then no remote-fault notification is detected on the given port at the time of inquiry.
Remote-fault count	The Remote-fault count displays the number of times the remote-fault notification is detected on a given interface. The number of times, include: • The time since the Interface Module was last powered on. • The time since the count was last cleared by the user. • The time since the interface was last configured as a LAN mode.
time last RFN detected	The time the remote-fault notification was last detected on a given interface.

Limits and restrictions

Current implementation with this feature has the following limitations:

- Works only on a 10GbE LAN interface. Information for ports in a WAN interface at the time of inquiry is not displayed. Information for a port that does not belong to 10 GbE module is not displayed.
- In a Management Module switchover state, the remote fault notification counts and detection time are maintained.
- The RFN counts of a port only reset to zero in the following conditions:
  - At slot initialization (power on).
  - When the **clear remote-fault** command is enabled on 10 GbE LAN interface.
  - When configuring a 10GbE port into WAN mode.

Local fault event detection and counters

Local fault event detection and counters are enabled for ports on 10GbE LAN physical interfaces when link fault signaling is enabled on the interface. If both local fault and remote fault events are detected on the same interface, then the remote fault event is reported.

If a port is down because of a local fault event, then a syslog message is generated to inform you of this event. The syslog message will display "(local fault)" in the dynamic log buffer. For more information on local fault syslog messages, refer to the *Brocade NetIron Administration Guide* . The local fault event is also indicated in the **show interface** command. In the **show interface** command, the reason is displayed as "(local fault)". For more information on the **show interface** command, refer to the *Brocade NetIron Administration Guide*.

Displaying and clearing local fault counters

To display local fault counters on 10GbE LAN physical interface, enter the following command.

The example above displays local fault counters for ports 2/1 and 2/2 in LAN physical mode.

```
device# show local-fault ethernet 2/1 to 2/2
Port
  Local Fault Detected
   Local-Fault Count
     time last Local Fault detected
---- --------------------  ------------------  ----------------------------
2/1
        yes
                 1
                 Apr  3 18:06:28
2/2
        yes
               1
                  Apr  3 18:06:28
```

To clear local fault counters on a 10 GbE LAN physical interface, enter the following command.

```
device# clear local ethernet 2/1 to 2/2
Local-fault stats for port 2/1 is cleared.
Local-fault stats for port 2/2 is cleared.
```

The example above displays ports 2/1 and 2/2 cleared for local-fault statistics.

Syntax: show or clear local [**ethernet slot#/port#** [**to slot#/port**] | **slot slot#**]

You can display information for local fault counters in a Brocade device by using the **show local-fault** command without options,

Or use the ethernet *<slot#/port#>* option to limit the display to a single ethernet port.

Or use the to *<slot#/port>* option for a range of ports.

Use the slot *<slot#>* option to limit the display to a single slot.

The following table describes the output of the **show local-fault** command.

TABLE 6 Display of show local-fault output

This field...	Displays...
Port	The *<port#>* variable specifies the port number for the interface module.
Local Fault Detected	The local-fault is detected on a given interface. If "Yes" is displayed, then the local-fault event is detected on the given port at the time of inquiry. If "No" is displayed, then no local-fault event is detected on the given port at the time of inquiry.
Local-fault count	The local-fault count displays the number of times the local-fault event is detected on a given interface. The number of times, include: • The time since the Interface Module was last powered on.

TABLE 6 Display of show local-fault output (continued)

This field...	Displays...
	• The time since the count was last cleared by the user. • The time since the interface was last configured as a LAN mode.
time last Local Fault detected	The time the local-fault event was last detected on a given interface.

Displaying BIP error information

The **show bip slot** command is used to display a table that contains the lane number for a Physical Coding Sublayer (PCS) lane and a count of Bit Interleaved Parity (BIP) errors for the specific PCS lane. The command output is provided for a lane where a counter is active. The output helps the user to identify the bit parity errors on each physical interface lane of the Brocade MLX 100 GbE modules.

The following example displays an output from the **show bip slot** command on the Brocade NetIron devices.

```
device# show bip slot 3
Port 3/1:
PCS Lane BIP Error Counters :
*****************************
Lane00 : 001 Lane01 : 001
Lane02 : 001 Lane03 : 001
Lane04 : 001 Lane05 : 001
Lane06 : 001 Lane07 : 001
Lane08 : 001 Lane09 : 001
Lane10 : 001 Lane11 : 001
Lane12 : 001 Lane13 : 001
Lane14 : 001 Lane15 : 001
Lane16 : 001 Lane17 : 001
Lane18 : 001 Lane19 : 001
Port 3/2:
PCS Lane BIP Error Counters :
*****************************
Lane00 : 000 Lane01 : 000
Lane02 : 000 Lane03 : 000
Lane04 : 000 Lane05 : 000
Lane06 : 000 Lane07 : 000
Lane08 : 000 Lane09 : 000
Lane10 : 000 Lane11 : 000
Lane12 : 000 Lane13 : 000
Lane14 : 000 Lane15 : 000
Lane16 : 000 Lane17 : 000
Lane18 : 000 Lane19 : 000
All show BIP done
```

NOTE

The BIP error counter is reset to zero when the command is run. When the counter reaches 255, it does not exceed 255. The counter is increased by a link going up or down and this is expected behavior.

Displaying Network Processor statistics

The Network Processor (NP) counters track the packets and bytes that enter the ingress NP and exit the egress NP. Counts displayed are since the last time the **clear np statistics** command was issued.

The **show np statistics command** displays the NP statistics for all interface modules within a device or for an interface in a specified slot or port. A routed packet drop counter is added to the **show np statistics** command. For more information on the routed packet drop counter, see Table 7. The following example displays an output from the **show np statistics** command on the Brocade NetIron XMR Series, Brocade NetIron CES Series and Brocade NetIron CER Series.

Output of the Brocade NetIron XMR Series is as follows.

```
device # show np statistics ethernet 10/4
NP STATs IPC reply from slot 10 length =1608
Port 10/4 RX
NP Rx Raw Good Packet          = (115458)
NP Rx Forward Packet            = (115458)
NP Rx Discard Packet           = (0)
NP Rx Unicast Packet           = (44571)
NP Rx Broadcast Packet         = (0)
NP Rx Multicast Packet         = (70887)
NP Rx Send to TM Packet        = (115458)
NP Rx Bad Packet               = (0)
NP Rx Lookup Unavailable       = (0)
NP Rx ACL Drop                 = (0)
NP Rx Priority 0/1 Drop        = (0)
NP Rx Priority 2/3 Drop        = (0)
NP Rx Priority 4/5 Drop        = (0)
NP Rx Priority 6/7 Drop        = (0)
NP Rx Suppress RPF Drop        = (0)
NP Rx RPF Drop                 = (0)
NP Rx IPv4 Packet              = (0)
NP Rx IPv6 Packet              = (0)
NP Rx Route-only Drop          = (0)
NP Rx IPv6 Suppress RPF Drop   = (0)
NP Rx IPv6 RPF Drop Count      = (0)
NP Rx IPv4 Byte                = (0)
NP Rx IPv6 Byte                = (0)
NP Rx Routed Packet Drop     = (0)
Port 10/4 TX
NP Tx Sent to MAC Packet       = (1365518)
NP Tx Raw Good Packet          = (1365518)
NP Tx Source Port Supress Drop = (0)
NP Tx Bad Packet Count         = (0)
NP Tx Unicast Packet           = (1324427)
NP Tx Broadcast Packet         = (1)
NP Tx Multicast Packet         = (41090)
NP Tx IPX HW Forwarded Packet  = (41090)
NP Tx Receive from TM          = (1365518)
NP Tx ACL Drop                 = (0)
NP Tx IPv4 Packet              = (0)
NP Tx IPv6 Packet              = (0)
NP Tx IPv4 Byte                = (0)
NP Tx IPv6 Byte                = (0)
```

Syntax: show np statistics [**ethernet slot/port**] [**slot slot-num**]

You can use the **ethernet** option and specify a *<slot/port>* variable to display NP statistics for an individual port.

You can use the slot option and specify a *<slot-num>* variable to display NP statistics for an individual interface module.

Output of the Brocade NetIron CES Series is as follows.

```
device#show np statistics
TD: Traffic Descriptor. Each TD has size of 512 Bytes
MODULE # 0 PPCR # 0 :
Ingress Counters :
Received packets                                = 0
Received TDs on traffic class 0                 = 0
Received TDs on traffic class 0                 = 0
Received TDs on traffic class 1                 = 0
Received TDs on traffic class 2                 = 0
Received TDs on traffic class 3                 = 0
Received TDs on traffic class 4                 = 0
Received TDs on traffic class 5                 = 0
Received TDs on traffic class 6                 = 0
Received TDs on traffic class 7                 = 0
Egress Counters :
Transmitted unicast packets                     = 0
Transmitted multicast packets                   = 0
Transmitted broadcast packets                   = 0
```

```
Filtered packets due to VLAN spanning tree          = 0
Tail dropped packets                                = 0
Control packets                                     = 0
Packets filtered due to egress forward restrictions = 0)
```

Syntax: show np statistics [**slot slot-num**]

You can use the **slot** option and specify a <slot-num> variable to display NP statistics for an individual interface module.

For Brocade NetIron CES Series and Brocade NetIron CER Series, you can either use **show np statistics** command or **show np statistics** [**slot** <slot-num>] command to display the NP statistics for an interface in a specified slot.

The **Tx** and **Rx** counters displayed are described in the following tables.

TABLE 7 Rx counters

Rx counter (per port)	Explanation
Rx Raw Good Packet	Number of good packets received from MAC
Rx Forward Packet	Number of forwarded packets by packet evaluation engine
Rx Discard Packet	Number of packets flagged for discard by packet evaluation engine
Rx Unicast Packet	Number of unicast (indicated by MAC DA) packets received
Rx Broadcast Packet	Number of broadcast (indicated by MAC DA) packets received
Rx Multicast Packets	Number of multicast (indicated by MAC DA) packets received
Rx Send to TM Packets	Number of packets sent to TM (= Rx Forward Packet - RL drops)
Rx Bad Packets	Number of packets that have MAC to NP interface errors
Rx Loopup Unavailable	Number of packets that have been dropped due to unavailability of the CAM interface for packet lookups
Rx ACL Drop	Drop counter for ACL drop on the ingress path
Rx Priority 0/1 Drop	Drop counter for ingress priority 0,1 packets
Rx Priority 2/3 Drop	Drop counter for ingress priority 2,3 packets
Rx Priority 4/5 Drop	Drop counter for ingress priority 4,5 packets
Rx Priority 6/7 Drop	Drop counter for ingress priority 6,7 packets
Rx Supress RPF Drop	Counter for suppressed RPF drops on the ingress path due to ACL override
Rx RPF Drop	Counter for RPF drop on the ingress
Rx IPv4 Packet	Raw packet count that have IPv4 EType (0x0800) and IP version of 0x4
Rx IPv6 Packet	Raw packet count that have IPv6 EType (0x86DD) and IP version of 0x6
Rx IPv6 Supress RPF Drop	Counter for IPv6 suppressed RFP drops on the ingress path due to ACL override
Rx IPv6 RPF Drop Count	Counter for IPv6 drop on the ingress
NP Rx Route-only Drop	Counts packets that have been dropped due to Route-Only configuration during MAC-DA processing.
Rx IPv4 Byte	Raw packet Bytes (+FCS) that have IPv4 etype (0x0800) and IP version equals 0x4
Rx IPv6 Byte	Raw packet Bytes (+FCS) that have IPv6 etype (0x86DD) and IP version equals 0x6
Rx Routed Packet Drop	Number of received IPv4 or IPv6 routed packets that are dropped because the TTL is 0, or because routing is not enabled on the given virtual interface.

TABLE 8 Tx counters

TX counter (per port)	Explanation
Tx Sent to MAC Packet	Total number of packets sent to MAC for transmit
Tx Raw Good Packet	Total number of packets sent to egress processing logic that pass the initial length checks (min, max, offsets, bad packet etc.)
Tx Source Port Suppression Drop	Number of packets dropped because of transmit source port suppression
Tx Bad Packet Count	Total number of packets dropped in egress logic that fail the initial length checks (min, max, bad packet etc.)
Tx Unicast Packet	Number of unicast packets transmitted (from MAC DA)
Tx Broadcast Packet	Number of broadcast packets transmitted (from MAC DA)
Tx Multicast Packet	Number of multicast packets transmitted (from MAC DA)
Tx Receive From TM	Number of packets received from TM
Tx ACL Drop	Number of packets that have been dropped by the Outbound ACL Logic
Tx IPv4 Packet	Number of IPv4 packets transmitted out the port (Etype==0x0800 & IPver == 0x4)
Tx IPv6 Packet	Number of IPv6 packets transmitted out the port (Etype==0x86DD & IPver == 0x6)
Tx IPv4 Byte	Counts packet Bytes (+FCS) that have IPv4 etype (0x0800) and IP version equals 0x4
Tx IPv6 Byte	Counts packet Bytes (+FCS) that have IPv6 etype (0x86DD) and IP version equals 0x6

Relationships between some counters

Some of the values for counters displayed using the **show np statistics** command are the result of adding the contents of more than one counter. Table 9 and Table 10 describe these relationships between NP counters displayed.

Total RX Packets	=	Rx Bad Packets + Rx Lookup Unavailable Packets + Rx Raw Good Packets
Rx Raw Good Packets	===	Rx Unicast Packets + Rx Multicast Packets + Rx Broadcast Packets+Rx IPv4 Packets + Rx IPv6 Packets + Rx Other Packets+Rx Forward Packets + Rx Discard Packets
Rx Forward Packets	=	Rx Sent to TM Packets + Rx RL drop packets
Rx Discard Packets	=	ACL drop + TTL drop + route-only drop + RPF drop + tag mismatch drop+ VLAN blocking drop + segment filtering drop+ drop by packet evaluation decisions +miscellaneous
Rx Priority Drops	=	RL drop + Rx Discard Packets

Tx Raw Good Packets	==	Tx Receive From TM Packets - Tx Bad Packets +Tx Unicast Packets + Tx Broadcast Packets + Tx Multicast Packets + Tx Source Port Suppression Drop
Tx Sent to MAC	==	Tx IPv4 Packets + Tx IPv6 Packets + Tx Others +Tx Raw Good Packets - Tx Source Port Suppression Drop - Tx ACL drop - Tx RL Drop - Tx Multicast TTL drop

Clearing the NP statistics counters

You can clear the NP statistics counters for an entire device or selectively by port or slot using the **clear np statistics** command as shown in the following.

```
device# clear np statistics
```

Syntax: clear np statistics [**ethernet slot/port**] [**slot slot-num**]

You can use the **ethernet** option and specify a *<slot/port>* variable to clear NP statistics for an individual port.

You can use the **slot** option and specify a *<slot-num>* variable to clear NP statistics for an individual interface module.

Enabling the Foundry Discovery Protocol and Reading Cisco Discovery Protocol Packets

This chapter discusses the following features:

- **Foundry Discovery Protocol (FDP)** - a protocol used by Brocade devices to advertise themselves to other Brocade devices.
- **Cisco Discovery Protocol (CDP)** - a protocol used by Cisco devices to advertise themselves to other Cisco devices. Brocade devices use this protocol to learn device and interface information for Cisco devices in the network.

  NOTE
  On platforms that support the Ethernet Service Instance (ESI) framework: FDP and CDP may be configured in the default ESI. FDP and CDP are not supported under user-defined ESIs.

Using FDP

FDP enables Brocade devices to advertise themselves to other Brocade devices on the network. When you enable FDP on a Brocade device, the device periodically advertises information including the following:

- Hostname (device ID)
- Product platform and capability
- Software version
- VLAN and Layer 3 protocol address information for the port sending the update.

A device running FDP sends FDP updates on Layer 2 to MAC address 01-E0-52-CC-CC-CC. Other devices listening on that address receive the updates and can display the information in the updates.

NOTE
FDP is disabled by default on Brocade devices. FDP must be enabled on devices that support FDP and can receive FDP updates from its neighbors.

Configuring FDP

The following sections describe how to enable FDP and how to change the FDP update and hold timers.

Enabling FDP globally

To enable a Brocade device to globally send FDP packets, enter the following command at the global CONFIG level of the CLI.

```
device(config)# fdp run
```

Syntax: [no] fdp run

The feature is disabled by default.

NOTE
If FDP is globally enabled on a Brocade device, all the interfaces, by default, have FDP enabled in it. In this case, the **show run** command does not display any running info about FDP configuration in its output.

Enabling FDP at the interface level

You can enable FDP at the interface level by entering the following commands.

```
device(config)# int e 2/1
device(config-if-e10000-2/1)# fdp enable
```

Syntax: [no] fdp enable

By default, the feature is enabled on all the interfaces once FDP is enabled on the device. It is not enables globally.

NOTE
To remove an interface from global FDP configuration, run the **no fdp enable** command in the interface mode explicitly. In this case, the **show run** command displays the running configuration information for the specific interface at that instance.

NOTE
FDP is not supported on VPLS/VLL endpoints. By removing FDP from the configuration, the **no fdp enable** will stay in the configuration.

Changing the FDP update timer

By default, a device enabled for FDP sends an FDP update every 60 seconds. You can change the update timer to a value from 5 - 900 seconds.

To change the FDP update timer, enter a command such as the following at the global CONFIG level of the CLI.

```
device(config)# fdp timer 120
```

Syntax: [no] fdp timer *secs*

The *secs* parameter specifies the number of seconds between updates and can be from 5 - 900 seconds. The default is 60 seconds.

Changing the FDP hold time

By default, a device that receives an FDP update holds the information until one of the following events occurs:

- The device receives a new update.
- 180 seconds have passed since receipt of the last update. This is the hold time.

Once either of these events occurs, the device discards the update.

To change the FDP hold time, enter a command such as the following at the global CONFIG level of the CLI.

```
device(config)# fdp holdtime 255
```

Syntax: [no] fdp holdtime *secs*

The *secs* parameter specifies the number of seconds a device that receives an FDP update can hold the update before discarding it. You can specify from 10 - 255 seconds. The default is 180 seconds.

Displaying FDP information

You can display the following FDP information:

- FDP entries for neighbors

- FDP entries for individual devices
- FDP information for an interface on the device you are managing
- FDP packet statistics

NOTE
If the device has intercepted CDP updates, the CDP information is also displayed.

Displaying neighbor information

To display a summary of all the neighbors that have sent FDP updates to this device, enter the following command.

```
deviceA# 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
   -------------- ------------ ------ ---------- ----------- -------------
   deviceB    Eth 2/9      178    Router     device Rou Eth 2/9
```

Syntax: show fdp neighbor [**ethernet** *slot/portnum*] [**detail**]

The **ethernet***slot* / *portnum* parameter lists the information only for neighbor information received on the specified interface.

The **detail** parameter lists detailed neighbor information received on all the interfaces for which FDP is enabled on the device.

The **show fdp neighbor** command, without optional parameters, displays the following information.

TABLE 11 Summary FDP and CDP neighbor information

This line...	Displays...
Device ID	The hostname of the neighbor.
Local Int	The interface on which this device received an FDP or CDP update for the neighbor.
Holdtm	The maximum number of seconds this device can keep the information received in the update before discarding it.
Capability	The role the neighbor is capable of playing in the network.
Platform	The product platform of the neighbor.
Port ID	The interface through which the neighbor sent the update.

To display detailed information, enter the following command.

```
device# show fdp neighbor detail
Device ID: NetIronB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: NetIron Router,  Capabilities: Router
Interface: Eth 2/9
Port ID (outgoing port): Eth 2/9 is TAGGED in following VLAN(s):
 9 10 11
Holdtime : 176 seconds
Version :
Brocade, 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 12 Detailed FDP and CDP neighbor information

This line...	Displays...
Device ID	The hostname of the neighbor. In addition, this line lists the VLAN memberships and other VLAN information for the neighbor port that sent the update to this device.
Entry address(es)	The Layer 3 protocol addresses configured on the neighbor port that sent the update to this device. If the neighbor is a Layer 2 Switch, this field lists the management IP address.
Platform	The product platform of the neighbor.
Capabilities	The role the neighbor is capable of playing in the network.
Interface	The interface on which this device received an FDP or CDP update for the neighbor.
Port ID	The interface through which the neighbor sent the update.
Holdtime	The maximum number of seconds this device can keep the information received in the update before discarding it.
Version	The software version running on the neighbor.

Displaying FDP entries

To display the detailed neighbor information for a specific device, enter a command such as the following.

```
device# show fdp entry deviceB
Device ID: deviceB configured as default VLAN1, tag-type8100
Entry address(es):
Platform: device Router,  Capabilities: Router
Interface: Eth 2/9
Port ID (outgoing port): Eth 2/9 is TAGGED in following VLAN(s):
 9 10 11
Holdtime : 176 seconds
Version :
Brocade, Inc. Router, IronWare Version 07.6.01b1T53 Compiled on Aug 29
2002 at 10:35:21 labeled as B2R07601b1
```

Syntax: show fdp entry * | *device-id*

The * | *device-id* parameter specifies the device ID. If you enter *, the detailed updates for all neighbor devices are displayed. If you enter a specific device ID, the update for that device is displayed. For information about the display, refer to Displaying neighbor information on page 65.

Displaying FDP information for an interface

To display FDP information for an interface, enter a command such as the following.

```
BrocadeA# 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 FDP information.

Displaying FDP and CDP statistics

To display FDP and CDP packet statistics, enter the following command.

```
deviceA# 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

> **NOTE**
> Internal errors may be seen if the FDP packet size exceeds a limit of 1496 bytes, due to the configuration of the interface. This will prevent the transmission of FDP packets on this interface but will not impact the ability to receive FDP packets.

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.

```
Brocade# 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.

```
Brocade# clear fdp counters
```

Syntax: clear fdp counters

Reading CDP packets

Cisco Discovery Protocol (CDP) is used by Cisco devices to advertise themselves to other Cisco devices. By default, a Cisco device or a non-Cisco device forwards these packets without examining their contents. You can configure a device to intercept and display the contents of CDP packets. This feature is useful for learning device and interface information for Cisco devices in the network.

Brocade devices support intercepting and interpreting CDP version 1 and 2 packets.

> **NOTE**
> The device can interpret only the information fields that are common to both CDP version 1 and CDP version 2.

NOTE
When you enable interception of CDP packets, the device drops the packets. As a result, Cisco devices will no longer receive the dropped packets.

Enabling interception of CDP packets globally

To enable the device to intercept and display CDP packets, enter the following command at the global CONFIG level of the CLI.

```
device(config)# cdp run
```

Syntax: [no] cdp run

The feature is disabled by default.

NOTE
If CDP is globally enabled on a Brocade device, all the interfaces, by default, have CDP enabled on it. In this case, the **show run** command does not display any running information about the CDP configuration in its output.

Enabling interception of CDP packets on an interface

You can disable and enable CDP at the interface level.

You can enter the following commands.

```
device(config)# int e 2/1
device(config-if-e10000-2/1)# cdp enable
```

Syntax: [no] cdp enable

By default, the feature is enabled on all the interfaces once CDP is enabled on the device.

NOTE
To remove an interface from the global CDP configuration, run the **no cdp enable** command in the interface mode explicitly. In this case, the **show run** command displays the running configuration information for the specific interface at that instance.

NOTE
CDP is not supported on VPLS/VLL endpoints. By removing CDP from the configuration, the **no cdp enable** will stay in the configuration.

Displaying CDP information

You can display the following CDP information:

- Cisco neighbors
- CDP entries for all Cisco neighbors or a specific neighbor
- CDP packet statistics

Displaying neighbors

To display the Cisco neighbors the device has learned from CDP packets, enter the following command.

```
device# 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.

```
device# show fdp neighbors detail
Device ID: Router
Entry address(es):
  IP address: 10.95.6.143
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.

```
device# show fdp neighbors ethernet 1/1
Device ID: Router
Entry address(es):
  IP address: 10.95.6.143
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.

```
device# show fdp entry *
Device ID: Router
Entry address(es):
  IP address: 10.95.6.143
Platform: cisco RSP4,  Capabilities: Router
Interface: Eth 1/1,  Port ID (outgoing port): FastEthernet5/0/0
Holdtime : 124 seconds
Version :
Cisco Internetwork Operating System Software
IOS (tm) RSP Software (RSP-JSV-M), Version 12.0(5)T1,  RELEASE SOFTWARE
(fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Thu 19-Aug-99 04:12 by cmong
```

Syntax: show fdp entry * | *device-id*

For example, to display CDP entries for a specific device, specify the device ID.

```
device# show fdp entry Router1
Device ID: Router1
Entry address(es):
  IP address: 10.95.6.143
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.

```
device# 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

> **NOTE**
> Internal errors may be seen if the FDP packet size exceeds a limit of 1496 bytes, due to the configuration of the interface. This will prevent the transmission of FDP packets on this interface but will not impact the ability to receive FDP packets.

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.

```
device# clear fdp table
```

Syntax: clear fdp table

To clear CDP statistics, enter the following command.

```
device# clear fdp counters
```

Syntax: clear fdp counters

Using a Redundant Management Module

You can install a redundant management module in slot M1 or M2 of a Brocade chassis. (By default, the system considers the module in slot M1 to be the active management module and the module in slot M2 to be the redundant, or standby module. If the active module becomes unavailable, the standby module automatically takes over management of the system.

This chapter describes the redundant management module, how it works with the active module, and how to configure and manage it.

How management module redundancy works

This section explains the following:

- How management module redundancy works under normal operating conditions
- Events that cause a standby management module to assume the role of the active module (switchover)
- System implications when a switchover occurs

Management module redundancy overview

When you apply power to a Brocade device with two management modules installed, by default, the management module in slot M1 becomes the active module and the module in slot M2 becomes the standby module. (You can change the default active slot from M1 to M2 using the **active-management** command. Refer to Changing the default active chassis slot on page 73.)

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

- The flash memory on the active management module
- An Auxiliary Flash card in an Auxiliary Flash slot on the active management module.

Once the modules boot, the system compares the flash code and system-config files on the standby module to the files on the active module. If the files are not the same, the files on the standby module are synchronized with those on the active module.

During normal operation, the active module handles tasks such as obtaining network topology and reachability information and determining the best paths to known destinations. The active module also monitors the standby module.

The standby module functions in an active standby mode. Configuration changes made from the CLI to the active management module are also written to the standby management module even if they are not written to flash memory. Synchronizing the system-config and running-config files on both modules allows the standby module to assume the role of active module seamlessly, if necessary.

The interface modules are not reset, and continue to forward traffic while the standby management module takes over operation of the system. The new now-active management module receives updates from the interface modules and sends verification information to the interface modules to ensure that they are synchronized. If the new active management module becomes out of sync with an interface module, information on the interface module may be overwritten, which can cause an interruption of traffic forwarding. An out of sync state should only occur if there is a layer 3 topology change elsewhere in the network during the management failover. Brocade devices support Layer 3 hitless failover with restart for high-availability routing in protocols such as BGP and OSPF. With these high-

availability features enabled, when a device experiences a failover or restart, forwarding disruptions are minimized, and route flapping diminished to provide continuous service.

Management module switchover

The following events cause the standby management module to become the active module, which is called a switchover :

- The active module becomes unavailable
- You perform a manual switchover
- You remove and replace the active management module

The following sections explain how the switchover occurs for each event.

Unavailable active module

The following events cause an active module to become unavailable and a switchover to occur:

- An active module experiences a problem significant enough to cause a reset of the module
- The active module loses power

Before a switchover occurs, the active module resets itself and sends an interrupt signal to the standby module. The standby module then becomes the active module and the interface modules continue to forward traffic.

The new active module begins to manage the system. When the original active module becomes available again or is replaced, it assumes the role of standby module.

Manual switchover

In some situations, you may want to manually switch the active module to the standby module. You can perform a manual switchover using the **switchover** command. For information about performing this task, refer to Manually switching over to the standby management module on page 76.

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

Removal and replacement of a management module

For information about how to remove and replace a management module, refer to "Replacing a Management Module" in the appropriate installation guide.

This section explains how management module redundancy is affected when you remove and replace an active or standby management module.

Removal and replacement of an active management module

If you remove the active management module, the standby module automatically assumes the active role. When you insert a replacement module in the slot from which the original active module was removed, the replacement module assumes the standby role. This module boots from a source specified for the active module, for example:

- The flash memory on the active management module
- An Auxiliary flash card installed in the active management module

When the replacement module boots, the system compares the flash code and system-config files on the standby module to the files on the active module. If differences exist, the files on the standby module are synchronized to match those on the active module.

Removal and replacement of a standby management module

You can remove a standby management module without causing a switchover to occur. The active module continues to function normally. When the new module is installed, it assumes the role of standby, and boots from a source specified for the active module, for example:

- The flash memory on the active management module
- An Auxiliary flash card installed in the active management module

The system compares the flash code and system-config files on the replacement module to the files on the active module. If differences exist, the files on the standby module are synchronized to match those on the active module.

Switchover implications

When a switchover occurs between the active and standby modules, the following areas may be affected:

- Management sessions
- Syslog and SNMP traps
- BGP Peer Notification - Described in the *Brocade NetIron Routing Configuration Guide*

The following sections explain the implications for these areas.

Management sessions

You can establish management sessions using the management port on the active management module. If a switchover occurs, the management port on the original active module shuts down and all open CLI, Web Management Interface, and Brocade Network Advisor sessions with that port close. You can open new sessions with the new active module, if this module has the same management port connections.

For example, if you were accessing the Web Management Interface through a PC connected to the original active management port, you can open a new session if a PC is connected to the new active management port. Open a new session using the same IP address you used before the switchover. (If a switchover occurs, the IP address you configured on the original active module is automatically assumed by the new active module.)

Syslog and SNMP traps

When a switchover occurs, the Brocade system sends a Syslog message to the local Syslog buffer and to the Syslog server, if you have configured the system to use one. The system also sends an SNMP trap to the receiver, if one is configured.

When system power is restored, or the system is reset normally, a cold start message and trap are sent. However, if the system is reset as the result of switchover to the standby management module, the system sends a warm start message and trap.

Management module redundancy configuration

Configuring management module redundancy consists of performing one optional task (changing the default active chassis slot) as described in the following section.

Changing the default active chassis slot

By default, the Brocade system considers the module installed in slot M1 to be the active management module. However, you can change the default active chassis slot to M2 using the **active-management** command.

The **active-management** command determines which management module will become active after a power cycle. By default, the top management module of the Brocade XMR 16000 and Brocade MLX-16 or the left management module of the Brocade XMR 4000,

Brocade XMR 8000, Brocade MLX-4 and Brocade MLX-8 become active after a power cycle. This information is stored in the chassis's backplane EPROM and not in the configuration file.

To change the default active chassis slot from the default state of M1 to M2, enter the following commands.

```
device(config)# redundancy
device(config-redundancy)# active-management mgmt-2
```

Syntax: active-management mgt-module

The **mgt-module** parameter specifies the management module, either mgmt-1 or mgmt-2.

NOTE
This configuration has no effect on the **reload** and **boot** commands. it only applies to the power cycle when both management modules are installed in a chassis.

Managing management module redundancy

You can perform the following management tasks related to management module redundancy for Brocade devices:

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

File synchronization between active and standby management modules

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

- **Flash code** - The flash code can include the following files:
  - monitor, which contains the Real Time Operating System (RTOS) for the management module
  - primary, which contains the primary Multi-Service IronWare image for the management module
  - secondary, which contains the secondary Multi-Service IronWare image for the management module

A BrocadeMulti-Service IronWare image contains layer 1 - 3 software used by the management module.

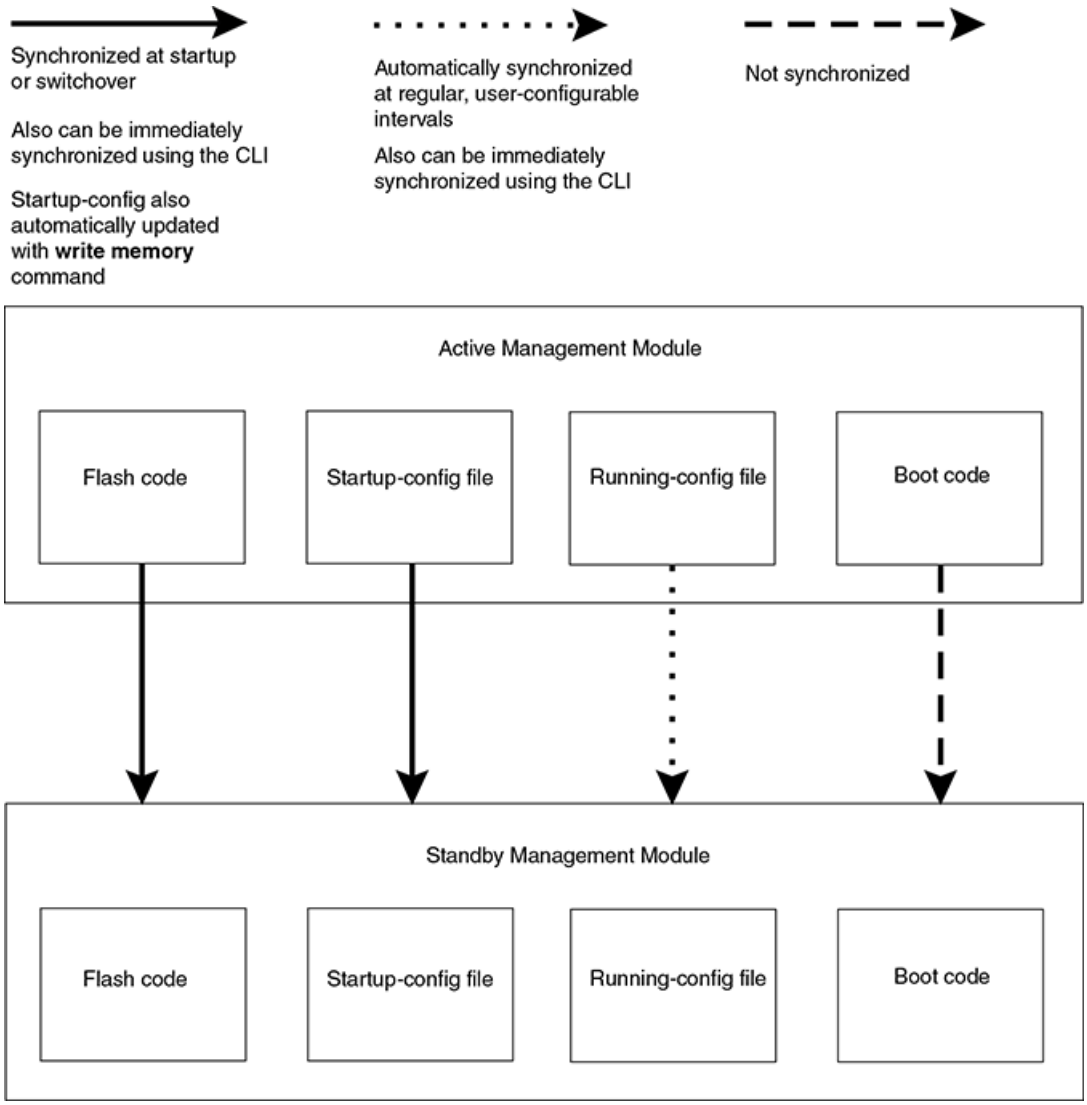
During startup or switchover, the flash code on the active module is compared to the flash code on the standby module. If the files differ, the files on the standby module are synchronized to the files on the active module. If you update the flash code on the active module, the flash code on the standby module is automatically synchronized (without comparison) to the new file on the active module.

- **System-config file** - The flash code includes the system-config file. During startup or switchover, the system-config file on the active module is compared to the system-config file on the standby module. If the files are different, the system-config file on the standby module is synchronized with that of the active module. When you save changes to the system-config file on the active module, the system-config file on the standby module is automatically (without comparison) synchronized to match the system-config file on the active module.
- **Running-config** - The running-config file resides in the Brocade system memory, and is automatically synchronized (without comparison) between the active and the standby module at regular intervals. The default interval is 7 seconds.
- **Boot code** - Each active and standby management module also includes boot code that is run when a module boots. The boot code resides in the boot flash of each module. Boot code is synchronized between the active and standby modules, which allows the system to use an older version of boot code on the standby module if desired.

NOTE
However, when the standby module is inserted to the standby slot, the images get synchronized to the standby image.

Figure 1 shows how the files are synchronized between the active module and the standby module.

FIGURE 1 Active and standby management module file synchronization



The Brocade system allows you to perform the following file synchronization tasks:

- Compare files on the active module with files on the standby module and immediately synchronize any files that are different.
- Immediately synchronize all files between the active and standby modules.

The following sections explain how to perform these tasks.

Comparing and synchronizing files

You can initiate a comparison of the flash code, system-config, and running-config files on the active management module with these files on the standby module and synchronize the files immediately if differences exist. When you synchronize the files, the active module files are copied to the standby module, replacing the standby module files.

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

```
device# sync-standby
```

Synchronizing files without comparison

You can synchronize the flash code, system-config file, and running-config file immediately without comparison. When you synchronize the files, active module files are copied to the standby module, replacing the files on the standby module.

To immediately synchronize the files between the active and standby modules, enter the following command at the Privileged EXEC level.

```
device# force-sync-standby
```

Manually switching over to the standby management module

You can cause the Brocade system to switch over to the standby module (and thus make it the active module). Enter the **switchover** command at the Privileged EXEC level.

```
device# switchover
```

In prior versions of the Multi-Service IronWare, typing the **switchover** command caused the Brocade device to switch control over to the redundant management module immediately without confirmation. Currently, you are presented with the question "**Are you sure?** " after the switchover command is executed. At this question, you can either type **y** to proceed with the switchover or type **n** to abort the switchover.

The following is an example of the new switchover procedure.

```
device#switchover
Are you sure? (enter 'y' or 'n'): y
```

> NOTE
> The switchover command should not be used immediately after downloading new code to the Brocade systems with redundant management modules.

Rebooting the active and standby management modules

You can reboot management modules, while maintaining the active and standby roles, using the **boot system** or **reload** commands. You can also reboot the standby module only, maintaining the standby role, using the **reboot-standby** command.

For example, to reboot the active and standby management modules from the primary BrocadeMulti-Service IronWare image in the management module flash memory, enter the following command at the Privileged EXEC level.

```
device# boot system flash primary
device# Are you sure? (enter 'y' or 'n'): y
```

Syntax: [no] boot system bootp | [**flash primary** | **flash secondary**] | **slot number filename** | **tftp ip-address filename**

The **flash primary** keyword specifies the primary BrocadeMulti-Service IronWare image in the management module flash memory. The **flash secondary** keyword specifies the secondary BrocadeMulti-Service IronWare image in the flash memory.

For the *number* parameter, specify 1 for Auxiliary Flash slot 1 on the active management module and 2 for Auxiliary Flash slot 2 on the active management module. For the *filename* parameter, specify the name of the image on the Auxiliary flash card.

The **tftp** keyword directs the Brocade device to boot from an BrocadeMulti-Service IronWare image on a TFTP server located at *ip-address* with the specified *filename*.

For example, to reboot the active and standby management modules, enter the following command at the Privileged EXEC level.

```
device# reload
```

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

```
device# reboot-standby
```

Monitoring management module redundancy

You can monitor the following aspects of management module redundancy:

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

The following sections explain how to monitor the management modules.

Determining management module status

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

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

Status LED

You can determine which management module is currently active and which is standby by observing the Active LED on each module. If this LED is on (green), the module is the active module. If this LED is off, the module is the standby module.

You can also observe the Pwr LED on each module. If this LED is on (green), the module is receiving power. If the LED is off, the module is not receiving power. (A module without power will not function as either the active or standby module.)

For information about what to do if these LED indicators are not what you expect, refer to the appropriate hardware installation guide.

Software

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

```
device# show module
       Module                              Status      Ports  Starting MAC
M1 (left): NI-XMR-MR Management Module     Active
M2 (right): NI-XMR-MR Management Module    Standby (Ready)
)
...
```

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

- **ACTIVE** - Current active management module
- **STANDBY** - Current standby management module.

The status of the standby module can be one of the following:

- **Init** - Currently initializing as the standby module
- **Ready** - Ready to take over as the active module, if necessary
- **Wait** - Waiting for boot information from the active management module
- **Sync** - Active module is currently synchronizing files on the standby module

Monitoring the status change of a module

The Brocade system now logs the status change of a module. The status change of a module is logged when the module becomes:

- **Up or Ready** - The module is running or ready to run.
- **Down** - The module is not running normally.

Upon the status change of a module, a message is logged in the syslog memory. At the CLI level, type the **show log** command to view the logged messages.

The following example displays a syslog message on an Interface Module in the Down state.

```
Feb  5 12:16:17:N:System: Module down in slot 1, reason REBOOTED. Error Code 0
```

The following example displays a syslog message on a Standby Management Module in the Down state.

```
Feb  5 14:38:58:N:System: Standby Management Module was down, reason Heartbeat Loss. Error Code 5
```

Displaying temperature information

All management, interface and switch fabric modules contain temperature sensors. By default, the Brocade system polls module temperature every 60 seconds. You can display the current temperature of the modules by entering either of the following commands:

- show chassis
- show temperature

For information about these commands, refer to the *appropriate hardware installation guide*.

Displaying switchover information

You can display the following information about a switchover:

- Redundancy parameter settings and statistics, including the number of switchovers that have occurred
- System log or traps logged on an SNMP trap receiver, including Information about whether a switchover has occurred.

To view the redundancy parameter settings and statistics, enter the following command at any level of the CLI.

```
device# show redundancy
=== MP Redundancy Settings ===
Default Active Slot = M1 (upper)
Running-Config Sync Period = 7 seconds
=== MP Redundancy Statistics ===
Current Active Session:
Active Slot=M2(lower),Standby Slot=M1(upper)(Ready State), Switchover Cause = No Switchover
Start Time = 1900-0-0  0:6:21 (Monday)
Previous Active Session #1:
Active Slot=M1(upper), Standby Slot=M2(lower), Switchover Cause = MP Upgrade to Ver3.7.0T163
Start Time = 1900-0-0  0:3:4 (Monday), End Time = 1900-0-0  0:6:21 (Monday)
Previous Active Session #2:
Active Slot = M2 (lower), Standby Slot = M1(upper), Switchover Cause = Active Rebooted
Start Time = 1900-0-0  0:1:1 (Monday), End Time = 1900-0-0  0:3:4 (Monday)
Previous Active Session #3:
Active Slot = M1 (upper), Standby Slot = M2(lower), Switchover Cause = MP Upgrade to Ver3.7.0T163
Start Time = 2036-2-6  6:43:54 (Wednesday), End Time = 1900-0-0  0:1:1 (Monday)
```

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

However, in three previous sessions, switchovers occurred. In sessions #1 and #3, the switchovers occurred because the software was upgraded to "Ver3.7.0T163". In session #2 the switchover occurred because the active module was rebooted. In sessions #1 and #3, the

modules installed in M1 were the active modules, while the modules installed in M2 were the standby modules. In session #2, the module installed in M2 was the active module, while the module installed in M1 was the standby module.

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

```
device# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 24 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Static Log Buffer:
Sep 28 11:31:25:A:Power Supply 1, 1st left, not installed
Sep 28 11:31:25:A:Power Supply 3, middle left, not installed
Sep 28 11:31:25:A:Power Supply 4, middle right, failed
Sep 28 11:31:25:A:Power Supply 5, 2nd right, not installed
Dynamic Log Buffer (50 lines):
Sep 27 18:06:58:I:Interface ethernet6/2, state up
Sep 27 18:06:57:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet6/2, state up
...
Sep 27 14:23:45:N:Module up in slot 6
Sep 27 14:23:45:N:Module up in slot 3
Sep 27 14:23:27:A:Management module at slot 9 state changed from standby to active
```

This output indicates that one switchover occurred.

Flash memory and auxiliary flash card file management commands

The Brocade system supports file systems in the following locations:

- Flash memory on the management module
- An Auxiliary flash card inserted in management module slots 1 or 2

Table 13 outlines the root directory for each file system.

TABLE 13 Brocade file system root directories

File system	Root directory
Flash memory	/flash/
Auxiliary flash card in slot 1	/slot1/
Auxiliary flash card in slot 2	/slot2/

This section describes commands that manage the files in flash memory and on the flash cards. Use the file management commands to perform the following tasks:

- Format a flash card
- Determine the current management focus
- Switch the management focus
- Display a directory of files
- Display the contents of a file
- Display the hexadecimal output of a file
- Create a subdirectory
- Remove a subdirectory
- Rename a file
- Change the read-write attribute of a file

- Delete a file
- Recover (undelete) a file
- Append one file to another (join two files)
- Perform copy operations using the **copy** command
- Perform copy operations using the **cp** command
- Load the system software from flash memory, a flash card, or other sources during system reboot
- Change the save location of the startup-config file from the default location (flash memory) to a flash card in slot 1 or 2

You can access all file management commands at the Privileged EXEC level of the CLI.

CAUTION
Do not add or remove a flash card while a file operation involving the slot where the flash card is installed is in progress. Doing so can result in corruption of the flash card. If this occurs, you may need to reformat the flash card to make it usable again. Reformatting erases all data stored on the card.

Verifying available flash space on the management module before an image is copied

The Management Module of the Brocade system accommodates 32 MB of flash space. However, as the size of the Interface Module, Management Module, and FPGA images increase, the Management Module flash may not have enough space to accommodate these images. The space in the Management Module flash is too small to hold more than two images (primary and secondary) and hence, downloading a new image is not possible without deleting one of the images that is already present in the flash.

Before an image is copied onto the Management Module or Interface Module, the software now checks to refer to if there is enough space available in the Management Module flash to support the copy operation. If there is not enough free space available on the Management Module flash, the following error message will display on the user interface.

The 32 MB flash space is capable of holding two Brocade NetIron CES Series or Brocade NetIron CER Series images (image size is about 11 MB). However, during the TFTP copy operation, it needs more buffer space. It is not possible to copy or update an existing image to a 32 MB flash, if there are two images in the flash already. If you try to copy or update an image, the following error message is displayed.

For TFTP copy operation, the following error message is displayed.

```
device#copy tftp flash 10.20.10.62 xmr04001b1.bin primary
There is not enough space on MP flash. Please clean up MP flash and retry, or use "delete-first" option.
TFTP: Download to primary flash failed - Flash is full
```

For SCP copy operation, the following error message is displayed.

```
C:\>scp xm04001b1.bin lab@10.22.2.21:image:primary
There is not enough space on MP flash. Please clean up MP flash and retry, or use "delete-first" option.
C:\>
```

In the example above the copy procedure is cancelled because there is not enough space on Management Module flash to copy the image. To make space for an image to be copied, you must clean up the flash space on the Management Module, and then retry copying the image again. You may also use the delete-first option, along with the CLI copy command, to make space for an image to be copied. The delete-first option allows you to delete existing target files on the Management Module flash.

The example below displays how the delete-first option is used. In this example, the existing secondary file image is removed from the flash to make space for a new image to be copied. The TFTP copy operation is able to successfully download the new image to the secondary flash.

```
device#copy tftp flash 10.53.1.82 xmr04001b1.bin secondary delete-first
Removing secondary from flash.
.....................................................................................TFTP: Download to
secondary flash done.
```

When the delete-first option is used, the existing target files are deleted only if there is enough free space to accommodate the copy operation. If, after the delete-first option is used and there is still a shortage of free space then the following error message will display.

```
device#copy tftp flash 10.53.1.82 xmr04001b1.bin secondary delete-first
There will not be enough space on MP flash even after deleting the target files. Please clean up MP flash
and retry.
```

Management focus

The **management focus** determines the default file system (flash memory or the flash card inserted in slot 1 or 2) to which a file management operation applies. When you power on or reload a Brocade system, by default, the management focus is on flash memory.

You can change the management focus from flash memory to a slot and subdirectory using the **cd** or **chdir** command. (For more information, refer to Switching the management focus on page 84.)

To determine the slot and subdirectory that have the current management focus, enter the **pwd** command. (For more information about this command, refer to Determining the current management focus on page 84.)

Most file management commands provide the option of specifying the file system to which the command applies. If you want the command to apply to the file system that has the current management focus, you do not need to specify the file system. If you want the operation to apply to the file system that does not have the current management focus, you must specify one of the following keywords:

- **flash** - indicates flash memory
- **slot1** - indicates the flash card inserted in slot 1
- **slot2** - indicates the flash card inserted in slot 2

For example, if you want to display a directory of files in flash memory and flash memory has the current management focus, you do not need to specify the **flash** keyword. However, if you want to display a directory of files for slot 1 and flash memory has the current focus, you must specify the **slot1** keyword.

Flash memory file system

The flash memory file system is flat, which means that it does not support subdirectories. As a result, you cannot create or delete subdirectories in this file system using the **md** /**mkdir** and **rd** /**rmdir** commands, respectively. Also, when specifying the syntax for the various file management commands, you will not need to specify a pathname to a subdirectory because it is not possible for a subdirectory to exist.

File naming conventions

A file name in the flash memory file system can contain a maximum of 31 characters. File name are case sensitive. The flash memory file system does not accept spaces as part of a file name.

The following characters are valid in file names:

- All upper and lowercase letters
- All digits
- Any of the following special characters:

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

Auxiliary flash card file system

The Auxiliary flash card file system is hierarchical, which means that it supports subdirectories. Therefore, you can create or delete subdirectories in this file system using the **md** /**mkdir** and **rd** /**rmdir** commands, respectively. Also, when specifying the syntax for the various file management commands, you may need to specify a pathname to a subdirectory as appropriate to manipulate a file in a subdirectory.

Auxiliary flash card subdirectories

The full path name for the location of a file can be a maximum of 256 characters. You can nest subdirectories as deep as you want as long as the full path name is 256 characters or less.

When you include a subdirectory path in a file management command, use a slash between each level. For example, to create a subdirectory for flash code and copy a flash image file to the subdirectory, enter commands such as the following.

```
device# mkdir slot1 /switchCode/initial-release
```

These commands create two levels of subdirectories on the flash card in Auxiliary flash slot 1.

File and subdirectory naming conventions

The Auxiliary flash slots supports file names of up to 32 characters. File names are not case sensitive. Thus, the software considers the name "test.cfg" and "TEST.CFG" to be the same.

Files and subdirectory names can be up to 32 characters long, including spaces and the special characters listed. The following characters are valid in file and subdirectory names:

- All upper and lowercase letters
- All digits
- Spaces
- Any of the following special characters:
  - $
  - %
  - '

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

You can use spaces in a file or subdirectory name if you enclose the name in double quotes. For example, to specify a subdirectory name that contains spaces, enter a string such as the following: "a long subdirectory name".

A subdirectory or file name can be a maximum of 256 characters long. A complete subdirectory path name cannot contain more than 256 characters.

There is no maximum file size. A file can be as large as the available flash card space.

NOTE
Auxiliary flash card file system applies to the Brocade NetIron XMR Series and Brocade NetIron MLX Series only.

Wildcards

Commands to display a directory of files, to change the read-write attribute of a file, or to delete files accept wildcards in the file name (*file-name*). With these commands, you can use "*" (asterisk) as a wildcard for any part of the name. For example, all the following values are valid for *file-name*:

- teststartup.cfg
- test*.cfg
- nmb02200.bin
- *.bin
- m*.bin
- m*.*

Formatting a flash card

The flash cards shipped with a management module are pre-formatted for the 16 FAT file system used by the modules. If you want to use a flash card that is not formatted for the 16 FAT file system, you need to reformat the flash card before you can store files on it.

CAUTION
Make sure the flash card is empty or does not contain files you want to keep. Formatting a flash card completely erases all files on the card.

CAUTION
Once you start the formatting process, you cannot stop it. Even if you enter CTRL-C to stop the CLI output and a new prompt appears, the formatting continues. Make sure you want to format the card before you enter the command.

To reformat a flash card in slot 2 on the management module, for example, enter the following command.

```
device# format slot2
.....................................................
.....................................................
.....................................................
.....................................
 80809984 bytes total card space.
 80809984 bytes available on card.
     2048 bytes in each allocation unit.
    39458 allocation units available on card.
```

Syntax: format slot1 | slot2

The **slot1 | slot2** keyword specifies the Auxiliary flash slot that contains the flash card you are formatting.

Determining the current management focus

For conceptual information about management focus, refer to Management focus on page 81.

To determine which file system has the current management focus, enter the following command.

```
device# pwd
Flash /flash/
```

In this example, the management focus is the flash memory.

In the following example, the management focus is the root directory of the flash card in slot 1.

```
device# pwd
/slot1/
```

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

```
device# pwd
/slot1/test/
```

Switching the management focus

The effect of file management commands depends on the file system that has the current management focus. For example, if you enter a command to delete a file and do not specify the location of the file, the software attempts to delete the file from the location that currently has the management focus.

By default, the management focus is on the flash memory on the management module. You can switch the focus from flash memory to flash cards in slot 1 or slot 2 on the management module using the **cd** or **chdir** commands, which have the same syntax and function exactly the same.

For example, to switch the focus from flash memory to the flash card in slot 2, enter the following command.

```
device# cd /slot2
device#
```

When you enter this command, the software changes the management focus to slot 2 then displays a new command prompt. If a slot you specify does not contain a flash card, the software displays the message shown in the following example.

```
device# cd /slot2
Device not present
```

Syntax: cd directory-pathname

Syntax: chdir directory-pathname

For the *directory-pathname* parameter for both **cd** and **chdir** commands, specify /slot1 or /slot2 to switch the focus to slot 1 or slot 2, respectively. Specify /flash to switch the focus to flash memory.

After you have switched the focus to slot 2, you can specify the *directory-pathname* parameter to switch the focus to a subdirectory on a flash card inserted in slot 2. For example, to switch the focus from the root directory level (/) of slot 2 to the subdirectory named "PLOOK," enter the following command.

```
device# cd /PLOOK
```

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

```
device# cd /PLOOK
Path not found
```

If you are certain the path you specified exists, make sure you are at the correct level to reach the path. For example, if you are already at the PLOOK level, the CLI cannot find the subdirectory "/PLOOK" because it is not a subdirectory from the level that currently has the management focus.

To change the management focus back to flash memory, enter the following command.

```
device# cd /flash
device#
```

Displaying a directory of the files

You can display a directory of the files in the flash memory on the management module, or on a flash card inserted in management module slot 1 or slot 2 using the **dir** or **ls** commands.

The software displays the directory of the file system that has the current management focus. By default, flash memory has the management focus. However, you do not need to change the focus to list the files on the file system that does not currently have management focus. In this case, you can specify the */path-name/* parameter with the **dir** or **ls** commands to display the directory of the desired file system.

For example, to display a directory of the files in flash memory, if flash memory has the management focus, enter the following command.

```
device# dir
Directory of /flash/
07/28/2003 15:57:45              3,077,697 1060.tmp
07/28/2003 15:56:10              3,077,697 14082.tmp
07/28/2003 16:00:08              3,077,697 2084.tmp
07/25/2003 18:00:23                292,701 boot
00/00/0    00:00:00                     12 boot.ini
07/28/2003 14:40:19                840,007 lp-primary-0
07/28/2003 15:18:18                840,007 lp-secondary-0
07/28/2003 09:56:16                391,524 monitor
07/28/2003 15:08:12              3,077,697 primary
07/28/2003 16:02:23                  1,757 startup-config
07/25/2003 18:02:14                  1,178 startup.sj2
07/28/2003 14:28:47                  1,662 startup.spa
07/26/2003 12:16:29                  1,141 startup.vso
07/25/2003 18:11:01                  1,008 startup.vsr
07/28/2003 09:40:54                  1,554 startup.vsrp.ospf
               15 File(s)       14,683,339 bytes
                0 Dir(s)        15,990,784 bytes free
```

Syntax: dir ls | [**path-name**]

You can enter either **dir** or **ls** for the command name.

Specify the *path-name* parameter to display the following:

- The files that match the value for a flash memory directory, or flash card directory/subdirectory you specify

- The files that match the value for a name you specify

For example, to list only files that contain a *.tmp suffix in flash memory, if flash memory is the current management focus, enter a command such as the following.

```
device# dir *.tmp
Directory of /flash/
07/28/2003 15:57:45              3,077,697 1060.tmp
07/28/2003 15:56:10              3,077,697 14082.tmp
07/28/2003 16:00:08              3,077,697 2084.tmp
                3 File(s)        9,292,701 bytes
                0 Dir(s)        15,990,784 bytes free
```

For example, to display a directory of the files on the flash card in slot 2, if flash memory has the management focus, enter the following command.

```
device# dir /slot2/
Directory of /slot2
/
08/01/2003 18:25:28              3,092,508 PRIMARY
08/01/2003 18:28:06              3,092,508 primary.1234
08/01/2003 18:28:24                389,696 MONITOR
08/01/2003 18:28:30                389,696 MONITOR1
08/01/2003 18:28:01                389,696 MONITOR2
08/01/2003 18:28:03                389,696 MONITOR3
08/01/2003 18:29:04                389,696 MONITOR4
08/01/2003 18:29:12   <DIR>                DIR1
08/01/2003 18:32:03                389,696 1234567890.12345
08/01/2003 18:32:08                389,696 123456.123
08/01/2003 18:32:11                389,696 123456.123
08/01/2003 18:32:14                389,696 123456.123
08/01/2003 18:32:17                389,696 123456.123
               12 File(s)       10,081,976 bytes
                1 Dir(s)       114,577,408 bytes free
```

The following information is displayed for each file.

TABLE 14 CLI display of directory information

This field...	Displays...
File date	The date on which the file was placed in the flash memory or card, if the device system clock is set.
Time of day	The time of day at which the file was placed in the flash memory or card, if the device system clock is set.
File size	The number of bytes in the file.
Read-write attribute	If you have set the read-write attribute of the file to read-only, "R" appears before the file name. If the read-write attribute of the file is read-write (the default), no value appears in this column. For information, refer to Changing the read-write attribute of a file on page 90.
File name	The file name.
Long file name	This field applies to files on a flash card only. The longer file name applies if the file was created on a PC and the name is longer than the 8.3 format.

The directory also lists the total number of files that match the parameters you specified, the total number of bytes used by all the files, and the number of bytes still free.

Displaying the contents of a file

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

The software displays the specified file in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to display the file in a file system that does not currently have management focus. In this case, you can specify the */directory/ path-name* parameter with the **more** command to display the file in the desired file system.

For example, to display the contents of a file in flash memory, if flash memory has the current management focus, enter a command such as the following.

```
device# more cfg.cfg
```

Syntax: more [**/directory/**] **file-name**

Use the *directory* parameter to specify a directory in a file system that does not have current management focus.

Use the *path-name* parameter to specify the file you want to display.

For example, to display the contents of a file on the flash card in slot 2, if flash memory has the current management focus, enter a command such as the following.

```
device# more /slot2/cfg.cfg
```

Displaying the hexadecimal output of a file

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

The software displays the hexadecimal output of a specified file in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to display the hexadecimal output of a file in a file system that does not currently have management focus. In this case, you can specify the */directory/ file-name* parameter with the **hd** command to display the output of the file in the desired file system.

For example, to display the hexadecimal output of a file in flash memory, if flash memory has the current management focus, enter the following command.

```
device# hd cfg.cfg
```

Syntax: [no] hd [**/directory/**] **file-name**

Use the *directory* parameter to specify a directory in a file system that does not have current management focus.

Use the *file-name* parameter to specify a file for which you want to display the hexadecimal output.

For example, to display the hexadecimal output of a file in a flash card inserted in slot 2, if flash memory has the current management focus, enter the following command.

```
device# hd /slot2/cfg.cfg
```

Creating a subdirectory

Create a subdirectory in the flash card file system using the **md** and **mkdir** commands, which have the same syntax and function exactly the same.

> NOTE
> You cannot create subdirectories in the flash memory file system. Therefore, the **md** and **mkdir** commands do not apply to the flash memory file system.

The software creates a subdirectory in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to create a subdirectory in a file system that does not currently have management focus. In this case, you can specify the **slot1** or **slot2** keyword with the **md** or **mkdir** command to create the subdirectory in the desired file system.

For example, to create a subdirectory on the flash card inserted in slot 2, if the flash memory has current management focus, enter a command such as the following.

```
device# mkdir slot2 TEST
```

Syntax: [no] md | mkdir [slot1 | slot2] dir-name

You can enter either **md** or **mkdir** for the command name.

Specify the **slot1** or **slot2** keyword to create a subdirectory on the flash card in slot 1 or slot 2, respectively. If you do not specify one of these parameters, the command applies to the file system that currently has the management focus.

The *dir-name* parameter specifies the subdirectory name. You can enter a name that contains any combination of the following characters. Do not enter a slash " / " in front of the name. Remember, a file name preceded by a slash represents the absolute path name (/flash, /slot1, or /slot2).

- All upper and lowercase letters
- All digits
- Spaces
- Any of the following special characters:
  - $
  - %
  - '
  - -
  - _
  - @
  - ~
  - `
  - !
  - (
  -)
  - {
  - }
  - ^
  - #
  - &

You can use spaces in a subdirectory name if you enclose the name in double quotes. For example, to specify a subdirectory name that contains spaces, enter a string such as the following: "a long subdirectory name".

A subdirectory name can be a maximum of 256 characters long. A complete subdirectory path name cannot contain more than 260 characters.

The name is not case sensitive. You can enter upper- or lowercase letters, however the CLI displays the name using uppercase letters.

To verify successful creation of the subdirectory, enter a command such as the following to change to the new subdirectory level.

```
device# chdir /slot2/TEST
Current directory of slot2 is: /TEST
```

For information about changing the directory using the **cd** and **chdir** commands, refer to Switching the management focus on page 84.

Removing a subdirectory

You can remove a subdirectory from the flash card file system using the **rd** and **rmdir** commands, which have the same syntax and function exactly the same.

NOTE
You cannot remove subdirectories from the flash memory file system. Therefore, the **rd** and **rmdir** commands do not apply to the flash memory file system.

NOTE
You can remove a subdirectory only if the subdirectory does not contain files or other subdirectories.

The software will remove a subdirectory from the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to remove a subdirectory from a file system that does not currently have management focus. In this case, you can specify the **slot1** or **slot2** keyword with the **rd** or **rmdir** command to remove the subdirectory from the desired file system.

For example, to remove a subdirectory from the flash card inserted in slot 2, if the flash memory has current management focus, enter a command such as the following.

```
device# rmdir slot2 TEST
```

Syntax: [no] rd rmdir | [**slot1** | **slot2**] **dir-name**

You can enter either **rd** or **rmdir** for the command name.

Specify the **slot1** or **slot2** keyword to remove a subdirectory on the flash card in slot 1 or slot 2, respectively. If you do not specify one of these parameters, the command applies to the file system that currently has the management focus.

The *dir-name* parameter specifies the subdirectory you want to delete. You can enter a path name if the subdirectory is not in the current directory.

If you receive a message such as the following, enter the **pwd** command to verify that the management focus is at the appropriate level of the directory tree.

```
device# rmdir TEST
rmdir /slot1/test/dir1/temp failed - File not found
```

For information about using the **pwd** command, refer to Determining the current management focus on page 84.

Renaming a file

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

The software renames the file in the file system that has the current management focus flash memory by default. However, you do not need to change the focus to rename the file in a file system that does not currently have management focus. In this case, you can specify the */directory/ old-file-name* /directory/ *new-file-name* parameter with the **rename** or **mv** command to rename the file in the desired file system.

For example, to rename a file in flash memory, if flash memory has the current management focus, enter a command such as the following.

```
device# rename oldname newname
```

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

Syntax: [no] rename mv | [**/directory/**] **old-file-name** [**/directory/**] **new-file-name**

You can enter either **rename** or **mv** for the command name.

The */directory/* parameter specifies a directory in a file system that does not have current management focus. When moving a file, the path must remain at the same directory level. You cannot rename the directory and the directory can be nested a maximum of five levels.

NOTE
Moving files up to the root directory is not supported.

The *old-file-name* parameter specifies the original filename that you want to change.

The *new-file-name* parameter specifies the new filename that you want to assign to the original file. The new filename must have an equal or greater number of characters than the old filename. The new filename cannot exceed 32 characters.

NOTE
A new filename with fewer characters than the old filename is not supported.

For example, to rename a file on the flash card inserted in slot 2, if flash memory has the current management focus, enter a command similar to the following.

```
device# rename /slot2/oldname /slot2/newname
```

Changing the read-write attribute of a file

You can specify the read-write attribute of a file on a flash card as follows:

- **Read-only** - You can display or copy the file but you cannot replace (copy over) or delete the file.
- **Read-write** - You can replace (copy over) or delete the file. This is the default.

NOTE
All files in flash memory are set to the read-write attribute, which cannot be changed. You cannot change this attribute. Therefore, the **attrib** command does not apply to the flash memory file system.

To determine the current setting of the read-write attribute for a file, use the **dir** command to list the directory information for the file. Files set to read-only are listed with "R" in front of the file name. For information about the **dir** command, refer to Displaying a directory of the files on page 85.

The software will change the read-write attribute of the file in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to change this file attribute in a file system that does not currently have management focus. In this case, you can specify the **slot1** or **slot2** keyword with the **attrib** command to change the attribute of the file in the desired file system.

For example, to change the attribute of a file in slot2 to read-only, if flash memory has the management focus, enter a command similar to the following.

```
device# attrib slot2 ro goodcfg.cfg
```

Syntax: [no] attrib [**slot1** | **slot2**] **ro** | **rw file-name**

Specify the **slot1** or **slot2** keyword to change the attribute of a file on the flash card in slot 1 or slot 2, respectively. If you do not specify one of these keywords, the command applies to the file system that currently has the management focus.

The **ro** parameter specifies that the attribute of the file is set to read-only. The **rw** parameter specifies that the attribute of the file is set to read-write.

The *file-name* parameter specifies the file for which to change the attribute.

For example, to change the attribute of all files on the flash card in slot 2 to read-only, if flash memory has the current management focus, enter a command similar to the following.

```
device# attrib slot2 ro *.*
```

Deleting a file

You can delete a file from flash memory or a flash card inserted in slot 1 or slot 2 on the management module using the **delete** or **rm** command.

NOTE
The **delete** or **rm** command deletes all files in a file system unless you explicitly specify the files you want to delete.

NOTE
The software does not support an undelete option for the flash memory file system. Be sure you really want to delete the file before you issue this command.

The software will delete the file in the file system that has the current management focus. By default, flash memory has the management focus. However, you do not need to change the focus to delete the file in a file system that does not currently have management focus. In this case, you can specify the */directory/file-name* parameter with the **delete** or **rm** command to delete the file in the desired file system.

For example, to delete a file in flash memory, if flash memory has the current management focus, enter a command similar to the following.

```
device# delete cfg.cfg
```

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

Syntax: delete rm | [**slot1** | **slot2**] [**directory**] [**file-name**]

You can enter either **delete** or **rm** for the command name.

Specify the **slot1** or **slot2** keywords to delete all files on the flash card in slot 1 or slot 2, respectively.

The *directory* parameter specifies the directory in a file system that does not have the current management focus.

The *file-name* parameter specifies the file that you want to delete.

For example, to delete all files with names that start with "test" from flash memory, if flash memory has the current management focus, enter a command similar to the following.

```
device# delete test*.*
```

For example, to delete all files on the flash card in slot 2, if flash memory has the current management focus, you can enter one of the following commands.

```
device# delete /slot2/
```

or

```
device# delete slot2
```

Recovering ("undeleting") a file

You can recover or undelete a file you have deleted from a flash card file system using the **undelete** command.

NOTE
You can not recover or undelete a file from the flash memory file system. Therefore, the **undelete** command does not apply to the flash memory file system.

The software will recover the file in the file system that has the current management focus (flash memory by default). If you want to recover a file in a file system that does not have the current management focus, you must switch the management focus to the desired file system using the **cd** command. For more information about switching the management focus, refer to Switching the management focus on page 84.

For example, to undelete a file on the flash card in slot 2, if flash memory has the current management focus, enter a command such as the following.

```
device# cd slot2
device# undelete
Undelete file ?RIMARY ? (enter y or n) :y
Input one character: P
File recovered successfully and named to PRIMARY
```

For each file that can be undeleted from the flash card in slot 2, the CLI displays the remaining name entry in the file directory and prompts you for the first character of the file name. You can enter any valid file name character. You do not need to enter the character that was used before in the deleted file name.

Once you enter a character and the CLI undeletes the file, the CLI continues with the next file that can be undeleted. For each file, specify "y" or "n", and specify a first character for the files that you select to undelete.

NOTE
When you delete a file from a flash card, the CLI leaves the file intact but removes the first letter in the file name from the file directory. However, if you save file changes or new files that use part of the space occupied by the deleted file, you cannot undelete the file. The **undelete** command lists only the files that can be undeleted.

To end the undelete process, enter CTRL + C.

Appending a file to another file

You can append a file in flash memory or on a flash card to the end of another file in one of these file systems.

The software will append one file to another in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to append one file to another in a file system that does not currently have management focus. In this case, you can specify the /*source-dir-path*/ or /*dest-dir-path*/ parameters with the **append** command to append one file to another in the desired file system.

To append one file to another in flash memory, if flash memory has the current management focus, enter a command similar to the following.

```
device# append newacls.cfg startup-config.cfg
```

Syntax: [no] append [**source-file-system dest-file-system**] [**/source-dir-path/**] **source-file-name** [**/dest-dir-path/**] **dest-file-name**

Specify the *source-file-system* and *dest-file-system* parameters when you are appending a file on one file system to a file on another file system.

The [/*source-dir-path*/] *source-file-name* parameter specifies the file you are appending to the end of another file. If the file is not located in the current subdirectory (the subdirectory that currently has the management focus), specify the subdirectory path in front of the file name.

The [/*dest-dir-path*/]*dest-file-name* parameter specifies the file to which you are appending the other file. If the file is not located in the current subdirectory, specify the subdirectory path in front of the file name.

For example, to append a file in the root directory of slot 1 to another file in a subdirectory of slot 2, enter a command similar to the following.

```
device# append slot1 slot2 newacls.cfg /TEST/startup-config.cfg
```

Copying files using the copy command

For information about copying files using the **copy** command while upgrading software images, refer to "Basic Tasks in the Software Upgrade Process" in the appropriate hardware installation guide.

You can perform the following additional copy operations using the **copy** command:

- Copy files from one flash card to the other
- Copy files between a flash card and the flash memory on the management module
- Copy software images between active and standby management modules
- Copy files from a management module to an interface module
- Copy Brocade Multi-Service IronWare management module images from flash memory to a TFTP server
- Copy files between a flash card and a TFTP server
- Copy a startup-config file between a flash card and flash memory on the management module
- Copy a startup-config file between flash memory on the management module and a TFTP server
- Copy the running-config to a flash card or a TFTP server
- Load a running-config from a flash card or TFTP server into the running-config on the device

  NOTE
  Since the copy options require you to explicitly specify the flash card, you can perform a copy regardless of which flash card is on the currently active management module.

Copying files from one flash card to the other

To copy a file from one flash card to the other, enter the following command.

```
device# copy slot1 slot2 sales.cfg
```

Syntax: copy from-card to-card [**/from-dir-path/**] **from-name** [**/to-dir-path/**] [**to-name**]

For the *from-card* and *to-card* parameters, you can specify **slot1** or **slot2**.

The command shown in the example copies a file from the flash card in slot 1 to the flash card in slot 2. In this case, the software uses the same name for the original file and for the copy. Optionally, you can specify a different file name for the copy.

Copying files between a flash card and flash memory

To copy a file from a flash card to the primary area in flash memory, enter a command similar to the following.

```
device# copy slot1 flash
nmpr02200.bin primary
```

Syntax: copy slot1 | **slot2 flash** [**/from-dir-path/**] **from-name monitor** | **primary** | **secondary**

To copy a file from flash memory to a flash card, enter a command similar to the following.

```
device# copy flash slot2
nmpr02200.bin primary
```

Syntax: copy flash slot1 | **slot2 source-name monitor** | **primary** | **secondary** | **startup-config** [**dest-name**]

The command in this example copies a BrocadeMulti-Service IronWare image file from the primary area in flash memory onto the flash card in slot 2. In this case, the software uses the same name for the source file and for the destination file. Optionally, you can specify a different file name for the destination file.

Copying software images between active and standby management modules

To copy the monitor image from flash memory of the active management module to flash memory of the standby module, enter the following command.

```
device# copy flash flash
monitor standby
```

To copy the BrocadeMulti-Service IronWare image from the secondary location in flash memory on the active management module to the primary location in flash memory, enter the following command.

```
device# copy flash flash
primary
```

Syntax: copy flash flash primary [**standby**]

Specify the optional **standby** keyword to copy the BrocadeMulti-Service IronWare image from the secondary location in flash memory on the active management module to the primary location in flash memory on the standby module.

To copy the BrocadeMulti-Service IronWare image from the primary location in flash memory on the active management module to the secondary location in flash memory on the active module, enter the following command.

```
device# copy flash flash
secondary
```

Syntax: copy flash flash secondary [**standby**]

Specify the optional **standby** keyword to copy the BrocadeMulti-Service IronWare image from the primary location in the flash memory on the active management module to the secondary location in the flash memory on the standby module.

Copying BrocadeMulti-Service IronWare images from flash memory to a TFTP Server

You can copy BrocadeMulti-Service IronWare images from the primary and secondary locations in flash memory on the management module to a TFTP server.

For example, to copy the BrocadeMulti-Service IronWare image in the secondary location in flash memory to a TFTP server, enter a command similar to the following.

```
device# copy flash tftp
10.10.10.1 secondary.bak secondary
```

Syntax: copy flash tftp ip-addr dest-file-name primary | **secondary**

Copying files between a flash card and a TFTP server

Use the following methods to copy files between a flash card and a TFTP server.

> NOTE
> The Brocade system must have network access to the TFTP server.

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

```
device# copy slot1 tftp 192.168.1.17 notes.txt
```

Syntax: copy slot1 | **slot2 tftp ip-addr** [**/from-dir-path/**] **source-file** [**dest-file**]

The command in this example copies a file from slot 1 to a TFTP server. In this case, the software uses the same name for the source file and for the destination file. Optionally, you can specify a different file name for the destination file.

To copy a software image from a TFTP server to a flash card, enter a command similar to the following.

```
device# copy tftp slot1 192.168.1.17
nmpr02200.bin primary
```

Syntax: copy tftp slot1 | slot2 ip-addr [/from-dir-path/] source-file path-name | monitor | primary | secondary

The command in this example copies the primary BrocadeMulti-Service IronWare image from a TFTP server to a flash card in slot 1.

Copying the startup-config file between a flash card and flash memory

Use the following methods to copy a startup-config file between flash memory and a flash card. By default, the Brocade device uses the startup-config in the primary area of flash memory when you boot or reload the device.

> NOTE
> The Brocade device cannot configure from a startup-config file on a flash card. You cannot boot or reload from a flash card.

To copy a startup-config file from a flash card to flash memory, enter a command similar to the following.

```
device# copy slot1 startup-config test2.cfg
```

Syntax: copy slot1 | slot2 startup-config [from-dir-path/file-name]

This command copies a startup configuration named test2.cfg from the flash card in slot 1 into the flash memory on the device. The next time you reboot or reload, the device uses the configuration information in test2.cfg.

To copy the startup-config file on the device from flash memory onto a flash card, enter a command similar to the following.

```
device# copy startup-config slot1 mfgtest.cfg
```

Syntax: copy startup-config slot1 | slot2 [/to-dir-path/] to-name

This command copies the startup configuration from the flash memory on the device to a flash card in slot 1 and names the file mfgtest.cfg.

Copying the startup-config file between flash memory and a TFTP server

Use the following methods to copy a startup-config between flash memory and a TFTP server to which the Brocade system has access. By default, the device configures from the startup-config in the primary area of flash memory when you boot or reload the device.

To copy the startup-config on the device from flash memory to a TFTP server, enter a command similar to the following.

```
device# copy startup-config tftp 10.10.10.1 /backups/startup.cfg
```

Syntax: copy startup-config tftp ip-addr [/to-dir-path] to-name

To copy a startup-config file from a TFTP server to flash memory, enter a command similar to the following.

```
device# copy tftp startup-config 10.10.10.1 test.cfg
```

Syntax: copy tftp startup-config ip-addr [/from-dir-path] from-name

Copying the running-config to a flash card or a TFTP server

Use the following method to copy the config file on the Brocade device to a flash card or a TFTP server. The running-config contains currently active configuration information for the device. When you copy the running-config to a flash card or TFTP server, you are making a copy of the current configuration, including any configuration changes you have not saved to the startup-config.

To copy the running configuration for the device into a file on a flash card, enter a command similar to the following.

```
device# copy running-config slot1 runip.1
```

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

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

```
device# copy running-config tftp 10.10.10.1 runip.1
```

Loading a running-config from a flash card or a TFTP server

Use the following method to load configuration commands into the active configuration for the Brocade device.

> NOTE
> A configuration file that you create must follow the same syntax rules as the startup-config the device creates. Refer to "Dynamic Configuration Loading" in the *appropriate hardware installation guide*.

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

```
device# copy slot2 running-config runacl.2
```

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

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

To copy a running-config from a TFTP server, enter a command similar to the following.

```
device# copy tftp running-config 10.10.10.1 run.cfg overwrite
```

Syntax: copy tftp running-config ip-addr [/from-dir-path/] from-name [overwrite]

This command copies a running-config from a TFTP server and overwrites the active configuration for the device.

> NOTE
> You cannot use the overwrite option from non-console sessions, as it will disconnect the session.

When a configuration file is loaded using the `copy tftp running-config` command, the following commands within the configuration file are supported.

- **isis metric command**
- **set-overload-bit command**
- **admin-group**
- **cspf-group**
- **bypass-lsp**

Copying files using the cp command

Use the **cp** command to do the following:

- Copy files from flash memory to flash memory
- Copy files from flash memory to a flash card or vice versa
- Copy files from one flash card to another flash card

The software will copy a file in a file system to another location in the file system that has the current management focus (flash memory by default). However, you do not need to change the focus to copy a file from one location to another in a file system that does not

currently have management focus. In this case, you can specify the */source-dir-path/* or */dest-dir-path/* parameters with the **cp** command to copy a file to or from a file system that does not have current management focus.

For example, to copy a file from flash memory, which has the current management focus, to flash memory, enter a command similar to the following.

```
device# cp primary primary2
```

For example, to copy a file from flash memory, which has the current management focus, to the flash card in slot 2, enter a command similar to the following.

```
device# cp new.cfg /slot2
/cfg/new.cfg
```

Syntax: cp [**source-dir-path**] **source-file-name** [**dest-dir-path**] **dest-file-name**

The *source-dir-path* parameter specifies the directory pathname of the source file. Specify this parameter if the source file is in a file system that does not have current management focus. The *source-file-name* specifies the name of the file you want to copy.

The *dest-dir-path* parameter specifies the directory pathname of the destination file. Specify this parameter if you want to copy the source file to a file system that does not have current management focus. The *dest-file-name* specifies the name of the file you copied to a new destination.

For example, to copy a file from a flash card in slot 2 to flash memory, which has current management focus, enter the following command.

```
device# cp /slot2
/cfg/new.cfg new.cfg
```

For example, to copy a file from a flash card in slot 1 to a flash card in slot 2, neither of which has current management focus, enter the following command.

```
device# cp /slot1/cfg/new.cfg /slot2
/cfg/new.cfg
```

Loading the software

By default, the management module loads an BrocadeMulti-Service IronWare image from the primary location in flash memory. You can change the BrocadeMulti-Service IronWare image source for the system to one of the following sources for a single reboot or for all future reboots:

- The secondary location in flash memory
- A flash card inserted in slot 1 or 2
- A TFTP server
- A BOOTP server

If you specify a source other than the primary location in flash memory and for some reason the source or the BrocadeMulti-Service IronWare image is unavailable, the system uses the primary location in flash memory as a default backup source.

Rebooting from the system

To use a source besides the Multi-Service IronWare image in the primary location in flash memory for a single reboot, enter a command similar to the following at the Privileged EXEC level of the CLI.

```
device# boot system slot1 /slot1/xmr03000.bin
```

The command in this example reboots the system using the image xmr03000.bin located on the flash card in slot 1. This example assumes that the flash card in slot 1 is not the management focus.

Syntax: boot system slot1 | slot2 [/dir-path/] file-name

The **slot1 | slot2** keywords specify the flash card slot.

The *file-name* parameter specifies the file name. If the file is in a subdirectory, specify the subdirectory path in front of the file name. If the file name you specify is not a full path name, the CLI assumes that the name (and path, if applicable) you enter are relative to the subdirectory that currently has the management focus.

> NOTE
> This command also is supported at the boot PROM.

For example, to reboot the system using the image xmr03000.bin on a TFTP server, enter a command similar to the following.

```
device# boot system tftp 10.10.10.1 xmr03000.bin
```

Syntax: boot system tftp ip-address file-name

The *ip-address* parameter specifies the address of the TFTP server on which the desired image resides.

The *file-name* parameter specifies the name of the BrocadeMulti-Service IronWare image on the TFTP server.

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

```
device# boot system flash secondary
device# Are you sure? (enter 'y' or 'n'): y
```

Syntax: boot system flash secondary

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

```
device# boot system bootp
```

Syntax: boot system bootp

Configuring the boot source for future reboots

To change the BrocadeMulti-Service IronWare image source from the primary location in flash memory to another source for future reboots, enter a command similar to the following at the global CONFIG level of the CLI.

```
device(config)# boot system slot1 xmr03000.bin
```

The command in this example sets Auxiliary flash slot 1 as the primary boot source for the Brocade device. When you reload the software or power cycle the device, the device will look for the BrocadeMulti-Service IronWare image on the flash card in slot 1.

Syntax: boot system slot1 file-name | slot2 file-name | flash secondary | tftp ip-address file-name | bootp

> NOTE
> The command syntax is the same for immediately reloading and for changing the primary source, except the *file-name* must be the full path name. You cannot specify a relative path name. If the first character in the path name is not a slash (/), the CLI treats the name you specify as relative to the root directory. How the device responds to the command depends on whether you enter the command at the Privileged EXEC level or the global CONFIG level.

If you enter multiple **boot system** commands at the global CONFIG level, the software places them in the running-config in the order you enter them, and saves them to the startup-config in the same order when you save the configuration. When you reload or power cycle the device, the device tries the boot sources in the order they appear in the startup-config and running-config.

Saving configuration changes

You can configure the Brocade system to save configuration changes to a startup-config in flash memory or on a flash card in slot 1 or 2.

Displaying the current location for saving configuration changes

Enter the following command at the Privileged EXEC level of the CLI to display the current save location for the startup-config.

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

Specifying the location for saving configuration changes

By default, when you save configuration changes, the changes are saved to the startup-config in flash memory.To change the save location to a flash card in slot 1 or 2, enter a command similar to the following.

```
device# locate startup-config slot1 router1.cfg
device# write memory
```

The first command in this example sets the device to save configuration changes to the file named "switch1.cfg" in the flash card in slot 1. The second command saves the running-config to the router1.cfg file on the flash card in slot 1.

> NOTE
> In this example, after you save the configuration changes using the **write memory** command, the router1.cfg file will include the command that designates slot 1 as the save location for configuration changes.

Syntax: locate startup-config [**slot1** | **slot2** | **flash-memory**] [**/dir-path-name/**] **file-name**

The **locate** command is used only for saving the startup-config file to a different location. But once after reload, the system always picks up the startup-config file from the flash memory.

The **slot1**, **slot2**, and **flash-memory** keywords specify the flash card in slot 1 or slot 2 or flash memory as the save location for configuration changes.

Specify the *dir-path-name* parameter if you want to save the configuration changes to a directory other than the root directory of a flash card file system.

The *file-name* parameter indicates the name of the saved configuration file.

To change the save location back to flash memory, enter a command similar to the following.

```
device# locate startup-config flash-memory router1.cfg
device# write memory
```

File management messages

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

TABLE 15 Flash card file management messages

This message...	Means...
File not found	You specified a file name that the software could not find. Verify the command you entered to make sure it matches the source and destination you intended for the file operation.
Current directory is: *dir-path*	You have successfully changed the management focus to the slot and subdirectory indicated by the message.
Path not found	You specified an invalid path.
There is not enough space on the card	The flash card does not have enough space to hold the file you are trying to copy to it.
Access is denied	You tried to copy or delete a file that has the read-only attribute.
A duplicate file name exists	You tried to rename a file using a name that is already in use by another file.

TABLE 15 Flash card file management messages (continued)

This message...	Means...
Fatal error, can not read or write media	A hardware error has occurred. One possible cause of this message is removing the flash card while a file operation involving the card was in progress.
There is sharing conflict between format command and other read/write operations	The flash card is currently undergoing formatting. This message also appears if you enter a command to format the card while the card is being accessed for another file operation.
Invalid DOS file name	A filename you entered contains an invalid character (for example, ":" or "\").
File recovered successfully and named *file-name*	A file you tried to recover was successfully recovered under the name indicated in the message

Configuring LLDP

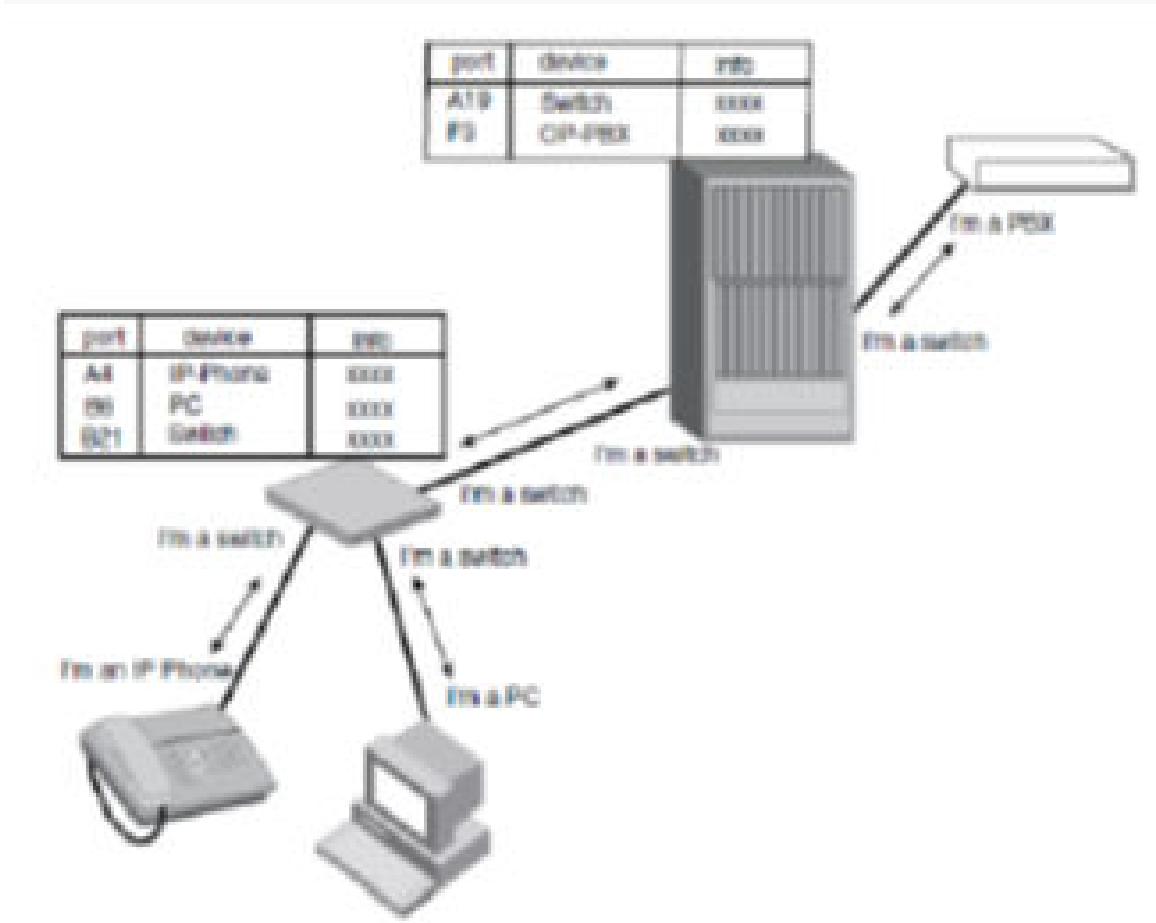
LLDP overview

LLDP enables a station attached to an IEEE 802 LAN or MAN to advertise its capabilities and to discover other stations in the same 802 LAN segments. The advertisements describe the network's physical topology and associated systems within that topology. For example, a station can advertise its management address, the address of the entities that manage the device, and the ID of the port to which the station is connected.

The information distributed through LLDP (the advertisement) is stored by the receiving device in a standard Management Information Base (MIB), accessible using a management protocol such as the Simple Network Management Protocol (SNMP). The information also can be viewed through the CLI, using **show LLDP** commands.

Figure 2 illustrates LLDP connectivity.

FIGURE 2 LLDP connectivity



General operating principles

LLDP use the services of the Data Link sub layers, Logical Link Control and Media Access Control, to transmit and receive information to and from other LLDP Agents (protocol entities that implement LLDP).

LLDP is a one-way protocol. An LLDP agent can transmit and receive information to and from another LLDP agent located on an adjacent device, but it cannot solicit information from another LLDP agent, nor can it acknowledge information received from another LLDP agent.

Operating modes

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. LLDP supports the following operating modes on physical interfaces:

- Transmit and Receive LLDP information. (System default)
- Transmit LLDP information only
- Receive LLDP information only

Transmit mode

An LLDP agent sends LLDP packets to adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

An LLDP agent initiates the transmission of LLDP packets whenever the transmit countdown timing counter expires, or whenever LLDP information has changed. When a transmit cycle is initiated, the LLDP manager extracts the MIB objects and formats this information into TLVs. The TLVs are inserted into an LLDPDU, addressing parameters are prepended to the LLDP packet, and the information is sent out LLDP-enabled ports to adjacent LLDP-enabled devices.

Receive mode

An LLDP agent receives LLDP packets from adjacent LLDP-enabled devices. The LLDP packets contain information about the transmitting device and port.

When an LLDP agent receives LLDP packets, it checks to ensure that the LLDP packets contain the correct sequence of mandatory TLVs, then validates optional TLVs. If the LLDP agent detects any errors in the LLDPDUs and TLVs, it drops them in software. TLVs that are not recognized but do not contain basic formatting errors, are assumed to be valid and are assigned a temporary identification index and stored for future possible alter retrieval by network management. All validated TLVs are stored in the neighbor database.

LLDP packets

LLDP agents transmit information about a sending device or port in packets called LLDP Data Units (LLDPDUs). All the LLDP information to be communicated by a device is contained within a single 1500 byte packet. LLDP information exceeding 1500 bytes will be truncated. A device receiving LLDP packets is not permitted to combine information from multiple packets.

Each LLDPDU consists of an untagged Ethernet header and a sequence of short, variable length information elements known as TLVs.

TLVs have Type, Length, and Value fields, where:

- **Type** identifies the kind of information being sent
- **Length** indicates the length (in octets) of the information string
- **Value** is the actual information being sent (for example, a binary bit map or an alpha-numeric string containing one or more fields).

TLV support

This section lists and describes LLDP TLV support.

LLDP TLVs

There are two types of LLDP TLVs, as specified in the IEEE 802.3AB standard:

- **Basic Management TLVs** consist of both optional general system information TLVs as well as mandatory TLVs.

Mandatory TLVs cannot be manually configured. They are always the first three TLVs in the LLDPDU, and are part of the packet header.

General system information TLVs are optional in LLDP implementations and are defined by the Network Administrator.

Brocade devices support the following Basic Management TLVs:

- – Chassis ID (mandatory)
  – Port ID (mandatory)
  – Time to Live (mandatory)
  – Port description
  – System name

  - System description
  - System capabilities
  - Management address
  - End of LLDPDU
- **Organizationally-specific TLVs** are optional in LLDP implementations and are defined and encoded by individual organizations or vendors. These TLVs include support for, but are not limited to, the IEEE 802.1 and 802.3 standards and the TIA-1057 standard.

Brocade devices support the following Organizationally-specific TLVs:

- 802.1 organizationally-specific TLVs
  - Port VLAN ID
  - VLAN name TLV
- 802.3 organizationally-specific TLVs
  - MAC/PHY configuration/status
  - Link aggregation
  - Maximum frame size

Mandatory TLVs

When an LLDP agent transmits LLDP packets to other agents in the same 802 LAN segments, the following mandatory TLVs are always included:

- Chassis ID
- Port ID
- Time to Live (TTL)

Chassis ID

The Chassis ID identifies the device that sent the LLDP packets.

There are several ways in which a device may be identified. A Chassis ID subtype, included in the TLV and shown in Table 16, indicates how the device is being referenced in the Chassis ID field.

TABLE 16 Chassis ID subtypes

ID Subtype	Description
0	Reserved
1	Chassis component
2	Interface alias
3	Port component
4	MAC address
5	Network address
6	Interface name
7	Locally assigned
8 - 255	Reserved

Brocade devices use Chassis ID subtype 4, the base MAC address of the device. Other third party devices may use a Chassis ID subtype other than 4. The Chassis ID will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Chassis ID (MAC address):  0012.f233.e2c0
```

The Chassis ID TLV is always the first TLV in the LLDPDU.

Port ID

The Port ID identifies the port from which LLDP packets were sent.

There are several ways in which a port may be identified, as shown in Table 17. A port ID subtype, included in the TLV, indicates how the port is being referenced in the Port ID field.

TABLE 17 Port ID subtypes

ID Subtype	Description
0	Reserved
1	Interface alias
2	Port component
3	MAC address
4	Network address
5	Interface name
6	Agent circuit ID
7	Locally assigned
8 - 255	Reserved

Brocade devices use port ID subtype 3, the permanent MAC address associated with the port. Other third party devices may use a port ID subtype other than 3. The port ID appears similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port ID (MAC address):  0012.f233.e2d3
```

TTL value

The Time to Live (TTL) Value is the length of time the receiving device should maintain the information acquired through LLDP in its MIB.

The TTL value is automatically computed based on the LLDP configuration settings. The TTL value will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**):

```
Time to live: 40 seconds
```

- If the TTL field has a value other than zero, the receiving LLDP agent is notified to completely replace all information associated with the LLDP agent or port with the information in the received LLDPDU.
- If the TTL field value is zero, the receiving LLDP agent is notified that all system information associated with the LLDP agent or port is to be deleted. This TLV may be used, for example, to signal that the sending port has initiated a port shutdown procedure.

Configuration considerations

- LLDP is supported on Ethernet interfaces only.
- If a port is 802.1X-enabled, the transmission and reception of LLDP packets will only take place while the port is authorized.
- Cisco Discovery Protocol (CDP) and Foundry Discovery Protocol (FDP) run independently of LLDP; therefore, these discovery protocols can run simultaneously on the same device.
- LLDP is supported on VPLS/VLL end-points and the behavior is the same as other interfaces.
- LLDP packets have the standard Multicast Destination MAC address and are sent with highest priority (7).
- By default, the Brocade device limits the number of neighbors per port to four (valid range is 1- 64), and staggers the transmission of LLDP packets on different ports, in order to minimize any high-usage spikes to the CPU.
- If the advertisements by the neighbor exceed the maximum value of the neighbor per port or if it exceeds the maximum neighbors configured at the global level then the new advertisements will be dropped.
- LLDP advertisements are limited to a single 1500 byte packet.

Using LLDP

LLDP is disabled by default on individual ports. To run LLDP, it must be enabled on a global basis (on the entire device).

Enabling LLDP

To enable LLDP globally, enter the **lldp run** command at the Global CONFIG level of the CLI.

```
device(config)# lldp run
```

Syntax: [no] lldp run

Changing the operating mode of a port

When LLDP is enabled on a global basis, by default, each port on the Brocade device will be capable of transmitting and receiving LLDP packets. Each port can be configured for a different operating mode on the Brocade device.

Configuring transmit and receive mode

To enable receipt and transmission of LLDP packets on individual ports, enter the**lldp enable ports ethernet** command at the Global CONFIG level of the CLI. The enabled ports are placed into transmit and receive mode by default.

```
device(config)# lldp enable ports ethernet 2/1
```

Syntax: [no] lldp enable ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Use the **[no]** form of the command to disable the receipt and transmission of LLDP packets on a port.

Configuring transmit mode

To change the LLDP operating mode from receive and transmit mode to transmit only mode, disable the transmit and receive mode, and enter the **lldp enable transmit ports ethernet** command.

```
device(config)# no lldp enable ports ethernet 2/4 2/5 2/6
device(config)# lldp enable transmit ports ethernet 2/4 2/5 2/6
```

```
The above command changes the LLDP operating mode on ports 2/4, 2/5, and 2/6 from transmit and receive mode
to transmit only mode.
```

Syntax: [no] lldp enable transmit ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Configuring receive mode

To change the LLDP operating mode from receive and transmit mode to receive only mode, disable the transmit and receive mode, and enter the **lldp enable receive ports ethernet** command at the Global CONFIG level of the CLI.

```
device(config)# no lldp enable ports ethernet 2/4
device(config)# lldp enable receive ports ethernet 2/4
The above command changes the LLDP operating mode on port 2/4 from transmit and receive mode to receive
only mode.
```

Syntax: [no] lldp enable receive ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Specifying the maximum number of LLDP neighbors

You can change the limit of the number of LLDP neighbors for which LLDP data will be retained, per device as well as per port.

Per device

To change the maximum number of neighbors for which LLDP data is retained for the entire system, use the **lldp max-total-neighbors** command. The default number of LLDP neighbors per device is 392.

```
device(config)# lldp max-total-neighbors 392
```

Syntax: [no] lldp max-total-neighbors *value*

The *value* variable specifies the total number of LLDP neighbors per device with a range of 16 to 8192.

Per port

To change the maximum number of LLDP neighbors for which LLDP data is retained for each port, use the **lldp max-neighbors-per-port** command. The default is number of LLDP neighbors per port is 4.

```
device(config)# lldp max-neighbors-per-port 4
```

Syntax: [no] lldp max-neighbors-per-port *value*

The *value* variable specifies the number of LLDP neighbors per port with a range of 1 to 64.

Enable bridging of LLDP BPDUs when LLDP not enabled

An interface which does not have LLDP enabled can be configured to bridge LLDP packets instead of dropping them. This action has to be specified explicitly by using the **forward-lldp** command.

NOTE
When LLDP is enabled this command will not have any effect on the behavior of LLDP. In other words, BPDUs will not be bridged.

The forward-lldp command must be issued on the physical port configuration, not in LAG configuration.

The LLDP BPDU forward command can be used at the interface level to allow bridging of LLDP BPDUs (LLDP BDPUs are normally dropped if LLDP is not configured on that interface).

```
device(config)# int e 2/1
device(config-if-e1000-1/2)#forward-lldp
```

Syntax: forward-lldp

Enabling LLDP SNMP notifications and Syslog messages

SNMP notifications and Syslog messages for LLDP provide data updates and general status.

When LLDP SNMP notifications are enabled, corresponding Syslog messages are enabled as well. When LLDP SNMP notifications are enabled, the device sends traps and corresponding Syslog messages whenever there are changes to the LLDP data received from neighboring devices.

LLDP SNMP notifications and corresponding Syslog messages are disabled by default. To enable SNMP notifications and Syslog messages on all interfaces, enter the **lldp enable snmp notifications ports all** command at the Global CONFIG level of the CLI.

```
device(config)# lldp enable snmp notifications ports all
```

Syntax: [no] lldp enable snmp notifications ports all

To enable or disable SNMP notifications and Syslog messages on a specific interface, enter the **lldp enable snmp notifications ports ethernet** command at the config level of the CLI.

```
device(config)# lldp enable snmp notifications ports ethernet  4/1
```

Syntax: [no] lldp enable snmp notifications ports ethernet *slot/port*

Specifying the minimum time between SNMP traps and Syslog messages

When SNMP notifications and Syslog messages for LLDP are enabled, the device sends no more than one SNMP notification and Syslog message within a 5 second period. You can adjust the amount of time between transmission of SNMP traps (lldpRemTablesChange) and Syslog messages from five seconds up 3600 seconds.

Use the **lldp snmp-notification-interval** command to change the amount of time between SNMP notifications.

```
device(config)# lldp snmp-notification-interval 5
```

Syntax: [no] lldp snmp-notification-interval *seconds*

The *seconds* variable specifies the notification interval with a range of 5 to 3600 seconds.

Changing the minimum time between LLDP transmissions

The LLDP transmit delay timer limits the number of LLDP frames an LLDP agent can send within a specified time frame. The LLDP transmit delay timer prevents an LLDP agent from transmitting a series of successive LLDP frames during a short time period, when rapid changes occur in LLDP. It also increases the probability that multiple changes, rather than single changes, will be reported in each LLDP frame.

When LLDP is enabled, the system automatically sets the LLDP transmit delay timer to the default of 2 seconds. To change the LLDP transmit delay timer setting, use the **lldp transmit-delay** command.

```
device(config)# lldp transmit-delay 2
```

Syntax: [no] lldp transmit-delay *seconds*

The *seconds* variable specifies the notification interval with a range of 1 to 8192 seconds.

> NOTE
> The LLDP transmit delay timer must not be greater than one quarter of the LLDP transmission interval (CLI command **lldp transmit-interval**).

Changing the interval between regular LLDP transmissions

The LLDP transmit interval specifies the number of seconds between regular LLDP packet transmissions. When LLDP is enabled, by default, the device waits 30 seconds between regular LLDP packet transmissions. To change the LLDP transmission interval, enter the **lldp transmit-interval** command.

```
device(config)# lldp transmit-interval 5
```

Syntax: [no] lldp transmit-interval *seconds*

The *seconds* variable specifies the notification interval with a range of 5 to 32768 seconds.

> NOTE
> Setting the transmit interval or transmit holdtime multiplier to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high. This in turn can affect how long a receiving device will retain the information if it is not refreshed.

Changing the holdtime multiplier for transmit TTL

The holdtime multiplier for transmit TTL is used to compute the actual time-to-live (TTL) value used in an LLDP frame. The TTL value is the length of time the receiving device should maintain the information. The default setting of holdtime multiplier for TTL to 4. This is the age out time for that particular advertisement. To compute the TTL value, the system multiplies the LLDP transmit interval by the holdtime multiplier. For example, if the LLDP transmit interval is 30 and the holdtime multiplier for TTL is 4, then the value 120 is encoded in the TTL field in the LLDP header.

To change the holdtime multiplier for TTL from the default value, use the **lldp transmit-hold** command.

```
device(config)# lldp transmit-hold 4
```

Syntax: lldp transmit-hold *value*

The *value* variable specifies holdtime multiplier for transmit TTL with a range of 4 to 10.

> NOTE
> Setting the transmit interval or transmit holdtime multiplier to inappropriate values can cause the LLDP agent to transmit LLDPDUs with TTL values that are excessively high.

Changing the minimum time between port reinitializations

The LLDP re-initialization delay timer specifies the minimum amount of time the device waits from when LLDP is disabled on a port, until it honors a request to re-enable LLDP on that port. When LLDP is enabled, the default is set to 2 seconds. The LLDP re-

initialization delay timer ensures that there is a defined minimum amount of time between successive LLDP frame transmission, thereby preventing a large number of LLDP frames to be sent at one time.

To change the LLDP re-initialization delay timer, enter the **lldp reinit-delay** command.

```
device(config)# lldp reinit-delay 2
```

Syntax: lldp reinit-delay *seconds*

The *seconds* variable specifies the LLDP re-initialization delay timer with a range of 1 to 10 seconds.

LLDP TLVs advertised by the Brocade device

When LLDP is enabled on a global basis, the Brocade device automatically advertises the following information, except as specified.

General system information

- Management address
- Port description
- System capabilities
- System description (not automatically advertised)
- System name

Management address

The management address is an IPv4 address that can be used to manage the device. If no management address is explicitly configured to be advertised, the Brocade device will use the first available IPv4 address configured on the following types of interfaces, in the following order of preference:

- Physical port on which LLDP will be transmitting the packet
- Loopback interface
- Virtual routing interface (VE)
- Router interface on a VLAN of which the port is a member
- Other physical interface

If no IP address is configured, the port's current MAC address will be advertised.

To advertise the IPv4 management address, enter the **lldp advertise management-address ipv4** command.

```
device(config)#lldp advertise management-address ipv4 10.157.2.1 ports e 1/4
```

The management address will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Management address (IPv4): 10.157.2.1
```

Syntax: [no] lldp advertise management-address ipv4 *ipv4address* **ports ethernet** *portlist* | **all**

ipv4 address is the address that may be used to reach higher layer entities to assist discovery by network management. In addition to the management address, the advertisement will include the system interface number and OID associated with the management address, if either or both are known.

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Port description

The port description TLV identifies the port from which the LLDP agent transmitted the advertisement. The port description is taken from the ifDescr MIB object from MIB-II.

By default, the port description is automatically advertised when LLDP is enabled on a global basis. To disable advertisement of the port description, enter the **lldp advertise port-description ports ethernet** command.

```
device(config)#no lldp advertise port-description ports e 2/4 to 2/12
```

The port description will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port description: "GigabitEthernet20"
```

Syntax: [no] lldp advertise port-description ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System capabilities

The system capabilities TLV identifies the primary functions of the device and indicates whether these primary functions are enabled. The primary functions can be one or more of the following:

- Repeater
- Bridge
- WLAN access point
- Router
- Telephone
- DOCSIS cable device
- Station only (devices that implement end station capability)
- Other

System capabilities for Brocade devices are based on the type of software image in use.

By default, the system capabilities are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter the **lldp advertise system-capabilities ports ethernet** command.

```
device(config)#no lldp advertise system-capabilities ports e 2/4 to 2/12
```

The system capabilities will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
System capabilities :   bridge
Enabled capabilities:   bridge
```

Syntax: [no] lldp advertise system-capabilities ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System description

The system description is the network entity. The information corresponds to the sysDescr MIB object. To advertise the system description, enter the **lldp advertise system-description ports ethernet** command.

```
device(config)#lldp advertise system-description ports e 2/4 to 2/12
```

The system description will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**):

```
device# show lldp local-info
Local port: 8/13
  + Chassis ID (MAC address): 0024.3891.1100
  + Port ID (MAC address): 0024.3891.125c
  + Time to live: 40 seconds
  + System name         : "IxANVL-1"
  + Port description    : "GigabitEthernet8/13"
  + System description  : "Brocade MLXe (System Mode: MLX), IronWare Version V\
                           5.3.0T163 Compiled on Jan  3 2012 at 18:01:00 label\
                           ed as V5.3.00b460"
  + System capabilities : bridge, router
    Enabled capabilities: bridge, router
  + 802.3 MAC/PHY          : auto-negotiation enabled
    Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                             100BaseTX-FD, 1000BaseT-HD, 1000BaseT-FD
    Operational MAU type   : 1000BaseT-FD
  + Link aggregation: not capable
  + Maximum frame size: 9216 octets
  + Port VLAN ID: 813
  + Management address (IPv4): 10.1.1.190
  + Management address (IPv4): 10.20.103.190
  + Management address (IPv6): 2001:DB8
  + Port-Protocol VLAN ID: not supported
```

Syntax: [no] lldp advertise system-description ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

System name

The system name is taken from the sysName MIB object. The sysName MIB object corresponds to the name defined with the CLI command **hostname** . By default, the system name is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter the **lldp advertise system-name ports ethernet** command.

```
device(config)#no lldp advertise system-name ports e 2/4 to 2/12
```

The system name will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
System name:  "NI"
```

Syntax: [no] lldp advertise system-name ports ethernet *slotnum/portnum* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.1 capabilities

Except for the VLAN name, the Brocade device will advertise the following 802.1 attributes when LLDP is enabled on a global basis:

- VLAN name (not automatically advertised)
- Untagged VLAN ID

VLAN name

The VLAN name TLV contains the name and VLAN ID of a VLAN configured on a port. An LLDPDU may include multiple instances of this TLV, each for a different VLAN.

To advertise the VLAN name, enter the **lldp advertise vlan-name vlan** command.

```
device(config)#lldp advertise vlan-name vlan 99 ports e 2/4 to 2/12
```

The VLAN name will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
VLAN name (VLAN 99): "Voice-VLAN-99"
```

Syntax: [no] lldp advertise vlan-name vlan *vlanID* **ports ethernet** *portlist* | **all**

For *vlan ID*, enter the VLAN ID to advertise.

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

Port and Protocol VLAN ID

The port and protocol VLAN TLV indicates if a port is capable of supporting port and protocol VLANs and whether it is enabled on the port. If port and protocol VLANs are enabled on the port, the advertisement also contains the port and protocol VLAN ID (PPVID). If the port is not capable of supporting port and protocol VLANs, or if the port is not enabled with any port and protocol VLAN, the PPVID number will be zero.

Use the **lldp advertise port-protocol-vlan-id ports ethernet** command to enable or disable advertising the port and protocol VLAN ID.

```
device(config)#lldp advertise port-protocol-vlan-id ports e 2/4 to 2/12
```

The port and protocol VLAN ID advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**):

```
Port-Protocol VLAN ID: not supported
```

Syntax: [no] lldp advertise port-protocol-vlan-id ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Untagged VLAN ID

The port VLAN ID TLV advertises the Port VLAN Identifier (PVID) that will be associated with untagged or priority-tagged frames. If the port is not an untagged member of any VLAN (that is, the port is strictly a tagged port), the value zero will indicate that.

By default, the port VLAN ID is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
device(config)#no lldp advertise port-vlan-id ports e 2/4 to 2/12
```

The untagged VLAN ID will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Port VLAN ID: 99
```

Syntax: [no] lldp advertise port-vlan-id ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword *to* to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

802.3 capabilities

Except for Power-via-MDI information, the Brocade device will advertise the following 802.3 attributes when LLDP is enabled on a global basis:

- Link aggregation information
- MAC/PHY configuration and status
- Maximum frame size

Link aggregation

Brocade devices advertise link aggregation information about standard link aggregation (LACP) as well as static trunk configuration. By default, link-aggregation information is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter the **lldp advertise link-aggregation ports ethernet** command.

```
device(config)#no lldp advertise link-aggregation ports e 2/12
```

Syntax: [no] lldp advertise link-aggregation ports ethernet *portlist* | **all**

The link aggregation advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Link aggregation: not capable
```

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.

MAC/PHY configuration status

The MAC/PHY configuration and status TLV includes the following information:

- Auto-negotiation capability and status
- Speed and duplex mode
- Flow control capabilities for auto-negotiation
- Port speed down-shift and maximum port speed advertisement
- If applicable, indicates if the above settings are the result of auto-negotiation during link initiation or of a manual set override action

By default, the MAC/PHY configuration and status information are automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter the **lldp advertise mac-phy-config-status ports ethernet** command.

```
device(config)#no lldp advertise mac-phy-config-status ports e 2/4 to 2/12
```

The MAC/PHY configuration advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
+ 802.3 MAC/PHY       : auto-negotiation enabled
   Advertised capabilities: 10baseT-HD, 10baseT-FD, 100baseTX-HD, 100baseTX-FD,
   fdxSPause, fdxBPause, 1000baseT-HD, 1000baseT-FD
   Operational MAU type: 100BaseTX-FD
```

Syntax: [no] lldp advertise mac-phy-config-status ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.Maximum frame size

Maximum frame size TLV

The maximum frame size TLV provides the maximum 802.3 frame size capability of the port. This value is expressed in octets and includes the four-octet Frame Check Sequence (FCS). The default maximum frame size is 1522. The advertised value may change depending on whether the **aggregated-vlan** or **jumbo** commands are in effect.

By default, the maximum frame size is automatically advertised when LLDP is enabled on a global basis. To disable this advertisement, enter a command such as the following.

```
device(config)#no lldp advertise max-frame-size ports e 2/4 to 2/12
```

The maximum frame size advertisement will appear similar to the following on the remote device, and in the CLI display output on the Brocade device (**show lldp local-info**).

```
Maximum frame size: 1522 octets
```

Syntax: [no] lldp advertise max-frame-size ports ethernet *portlist* | **all**

For *port list*, specify the ports in the format [*slotnum*/]*portnum*, where *slotnum* is required on chassis devices only. You can list all of the ports individually, use the keyword **to** to specify a range of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Note that using the keyword **all** may cause undesirable effects on some ports. The configuration will be applied to all ports, however, the ports that are not members of any VLAN will not send VLAN name advertisements.Maximum frame size

Displaying LLDP statistics and configuration settings

You can use the following CLI **show** commands to display information about LLDP settings and statistics:

- **show lldp** - Displays a summary of the LLDP configuration settings.
- **show lldp statistics** - Displays LLDP global and per-port statistics.
- **show lldp neighbors** - Displays a list of the current LLDP neighbors.
- **show lldp neighbors detail** - Displays the details of the latest advertisements received from LLDP neighbors.
- **show lldp local-info** - Displays the details of the LLDP advertisements that will be transmitted on each port.

LLDP configuration summary

To display a summary of the LLDP configuration settings on the device, enter the **show lldp** command at any level of the CLI.

The following shows an example report.

```
device#show lldp
LLDP transmit interval           : 10 seconds
LLDP transmit hold multiplier    : 4  (transmit TTL: 40 seconds)
LLDP transmit delay              : 1 seconds
LLDP SNMP notification interval  : 5 seconds
LLDP reinitialize delay          : 1 seconds
LLDP maximum neighbors           : 392
LLDP maximum neighbors per port  : 4
```

Syntax: show lldp

Table 18 describes the information displayed by the **show lldp statistics** command.

TABLE 18 Show lldp statistics

This field...	Displays...
LLDP transmit interval	The number of seconds between regular LLDP packet transmissions.
LLDP transmit hold multiplier	The multiplier used to compute the actual time-to-live (TTL) value of an LLDP advertisement. The TTL value is the transmit interval multiplied by the transmit hold multiplier.
LLDP transmit delay	The number of seconds the LLDP agent will wait after transmitting an LLDP frame and before transmitting another LLDP frame.
LLDP reinitialize delay	The minimum number of seconds the device will wait from when LLDP is disabled on a port, until a request to re-enable LLDP on that port will be honored.
LLDP maximum neighbors	The maximum number of LLDP neighbors for which LLDP data will be retained, per device.
LLDP maximum neighbors per port	The maximum number of LLDP neighbors for which LLDP data will be retained, per port.

LLDP statistics

The **show lldp statistics** command displays an overview of LLDP neighbor detection on the device, as well as packet counters and protocol statistics. The statistics are displayed on a global and per-port basis.

The following shows an example report.

```
device#show lldp statistics
Last neighbor change time: 23 hours 50 minutes 40 seconds ago
Neighbor entries added          : 14
Neighbor entries deleted        : 5
Neighbor entries aged out       : 4
Neighbor advertisements dropped : 0
Port       Tx Pkts    Rx Pkts    Rx Pkts    Rx Pkts    Rx TLVs    Rx TLVs Neighbors
             Total      Total  w/Errors Discarded Unrecognz Discarded  Aged Out
1            60963      75179         0          0          0          0         4
2                0          0         0          0          0          0         0
3            60963      60963         0          0          0          0         0
4            60963     121925         0          0          0          0         0
5                0          0         0          0          0          0         0
6                0          0         0          0          0          0         0
7                0          0         0          0          0          0         0
8                0          0         0          0          0          0         0
9                0          0         0          0          0          0         0
10           60974          0         0          0          0          0         0
11               0          0         0          0          0          0         0
12               0          0         0          0          0          0         0
13               0          0         0          0          0          0         0
14               0          0         0          0          0          0         0
```

Syntax: show lldp statistics

NOTE

You can reset LLDP statistics using the CLI command **clear LLDP statistics** . Refer to Resetting LLDP statistics on page 120.

NOTE

LLDP statistics are not preserved in the event of a module switchover.

Table 19 describes the information displayed by the **show lldp statistics** command.

TABLE 19 Show lldp statistics

This field...	Displays...
Last neighbor change time	The elapsed time (in hours, minutes, and seconds) since a neighbor last advertised information. For example, the elapsed time since a neighbor was last added, deleted, or its advertised information changed.
Neighbor entries added	The number of new LLDP neighbors detected since the last reboot or since the last time the **clear lldp statistics** all command was issued. This number includes the number entries added after timing out or aging out.
Neighbor entries deleted	The number of LLDP neighbors deleted since the last reboot or since the last time the **clear lldp statistics all** command was issued. This number includes the number of entries deleted after timing out or aging out.
Neighbor entries aged out	The number of LLDP neighbors dropped on all ports after the time-to-live expired. Note that LLDP entries age out naturally when a port's cable or module is disconnected or when a port becomes disabled. However, if a disabled port is re-enabled, the system will delete the old LLDP entries.
Neighbor advertisements dropped	The number of valid LLDP neighbors the device detected, but could not add. This can occur, for example, when a new neighbor is detected and the device is already supporting the maximum number of neighbors possible. This can also occur when an LLDPDU is missing a mandatory TLV or is not formatted correctly.
Port	The local port number.
Tx Pkts Total	The number of LLDP packets the port transmitted.
Rx Pkts Total	The number of LLDP packets the port received.

TABLE 19 Show lldp statistics (continued)

This field...	Displays...
Rx Pkts w/Errors	The number of LLDP packets the port received that have one or more detectable errors.
Rx Pkts Discarded	The number of LLDP packets the port received then discarded.
Rx TLVs Unrecognz	The number of TLVs the port received that were not recognized by the LLDP local agent. Unrecognized TLVs are retained by the system and can be viewed in the output of the **show LLDP neighbors** detail command or retrieved through SNMP.
Rx TLVs Discarded	The number of TLVs the port received then discarded.
Neighbors Aged Out	The number of times a neighbor's information was deleted because its TTL timer expired.

LLDP neighbors

The **show lldp neighbors** command displays a list of the current LLDP neighbors per port.

The following shows an example report.

```
device#show lldp neighbors
Lcl Port Chassis ID      Port ID         Port Description        System Name
1        0004.1234.0fc0  0004.1234.0fc0  GigabitEthernet9/1      BigIron RX 32~
1        00e0.5201.4000  00e0.5201.4000  GigabitEthernet0/1/1    BigIron RX 4~
3        00e0.5211.0200  00e0.5211.0203  GigabitEthernet4        BigIron RX 4~
4        00e0.5211.0200  00e0.5211.0202  GigabitEthernet3        BigIron RX 16~
4        00e0.5211.0200  00e0.5211.0210  GigabitEthernet17       BigIron RX 4~
15       00e0.5211.0200  00e0.5211.020f  GigabitEthernet16       BigIron RX 8~
16       00e0.5211.0200  00e0.5211.020e  GigabitEthernet15       BigIron RX 16~
17       00e0.5211.0200  00e0.5211.0211  GigabitEthernet18       BigIron RX 4~
18       00e0.5211.0200  00e0.5211.0210  GigabitEthernet17       BigIron RX 4~
```

Syntax: show lldp neighbors

The following table describes the information displayed by the **show lldp neighbors** command.

This field...	Displays...
Lcl Port	The local LLDP port number.
Chassis ID	The identifier for the device. Brocade devices use the base MAC address of the device as the Chassis ID.
Port ID	The identifier for the port. Brocade devices use the permanent MAC address associated with the port as the port ID.
Port Description	The description for the port. Brocade devices use the ifDescr MIB object from MIB-II as the port description.
System Name	The administratively-assigned name for the system. Brocade devices use the sysName MIB object from MIB-II, which corresponds to the CLI **hostname** command setting. **NOTE:** A tilde (~) at the end of a line indicates that the value in the field is too long to display in full and is truncated.

LLDP neighbors detail

The **show lldp neighbors detail** command displays the LLDP advertisements received from LLDP neighbors.

The following shows an example **show lldp neighbors detail** report.

> NOTE
> The **show lldp neighbors detail** output will vary depending on the data received. Also, values that are not recognized or do not have a recognizable format, may be displayed in hexadecimal binary form.

```
device#show lldp neighbors detail ports e 8/13
Local port: 8/13
  + Chassis ID (MAC address): 0024.3891.1100
  + Port ID (MAC address): 0024.3891.125c
  + Time to live: 40 seconds
  + System name         : "IxANVL-1"
  + Port description    : "GigabitEthernet8/13"
  + System description  : "Brocade MLXe (System Mode: MLX), IronWare Version V\
                           5.3.0T163 Compiled on Jan  3 2012 at 18:01:00 label\
                           ed as V5.3.00b460"
  + System capabilities : bridge, router
    Enabled capabilities: bridge, router
  + 802.3 MAC/PHY          : auto-negotiation enabled
    Advertised capabilities: 10BaseT-HD, 10BaseT-FD, 100BaseTX-HD,
                             100BaseTX-FD, 1000BaseT-HD, 1000BaseT-FD
    Operational MAU type   : 1000BaseT-FD
  + Link aggregation: not capable
  + Maximum frame size: 9216 octets
  + Port VLAN ID: 813
  + Management address (IPv4): 10.1.1.190
  + Management address (IPv4): 10.20.103.190
  + Management address (IPv6): 2001:DB8
  + Port-Protocol VLAN ID: not supported
Local port: 8/23
  + Chassis ID (MAC address): 0024.3891.1100
  + Port ID (MAC address): 0024.3891.1266
  + Time to live: 40 seconds
  + System name         : "IxANVL-1"
  + System description  : "Brocade MLXe (System Mode: MLX), IronWare Version V\
                           5.3.0T163 Compiled on Jan  3 2012 at 18:01:00 label\
                           ed as V5.3.00b460"
  + Port VLAN ID: 1
  + Management address (IPv4): 10.1.1.190
  + Management address (IPv4): 10.20.103.190
  + Management address (IPv6): 2001:DB8
```

A backslash (\) at the end of a line indicates that the text continues on the next line.

Except for the Neighbor field, the fields in the above output are described in the individual TLV advertisement sections in this chapter.

This field...	Displays...
Neighbor	The source MAC address from which the packet was received, and the remaining TTL for the neighbor entry.

Syntax: show lldp neighbors detail [**ports etherne t** *slotnum/portnum* | **all**]

If you do not specify any ports or use the keyword **all**, by default, the report will show the LLDP neighbor details for all ports.

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

LLDP configuration details

The **show lldp local-info** command displays the local information advertisements (TLVs) that will be transmitted by the LLDP agent.

NOTE

The **show lldp local-info** output will vary based on LLDP configuration settings.

The following shows an example report.

```
device#show lldp local-info ports ethernet 1/40
Local port: 1/40
  + Chassis ID (MAC address): 001b.edb3.f180
  + Port ID (MAC address): 001b.edb3.f1a8
  + Time to live: 40 seconds
  + System name          : "CES-151"
  + Port description     : "GigabitEthernet1/40"
  + System description   : "Brocade NetIron CES, IronWare Version V5.3.0T183 Co\
                           mpiled on Jan 03 2012 at 18:18:17 labeled as V5.3.0\
                           0b460"
  + System capabilities : bridge, router
    Enabled capabilities: bridge, router
  + 802.3 MAC/PHY          : auto-negotiation enabled
    Advertised capabilities: 1000BaseX-FD
    Operational MAU type   : 1000BaseT-FD
  + Link aggregation: aggregated (aggregated port ifIndex: 3)
  + Maximum frame size: 9216 octets
  + Port VLAN ID: 1
  + Management address (IPv4): 10.1.1.151
  + Management address (IPv4): 10.20.103.151
  + Management address (IPv6): 2001:DB8
  + Port-Protocol VLAN ID: not supported
```

A backslash (****) at the end of a line indicates that the text continues on the next line.

The fields in the above output are described in the individual TLV advertisement sections in this chapter.

Syntax: show lldp local-info [**ports ethernet** *slot num/portnum* | **all**]

If you do not specify any ports or use the keyword **all**, by default, the report will show the local information advertisements for all ports.

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Resetting LLDP statistics

To reset LLDP statistics, enter the **clear lldp statistics** command at the Global CONFIG level of the CLI. The Brocade device will clear the global and per-port LLDP neighbor statistics on the device (refer to LLDP statistics on page 116).

```
device#clear lldp statistics
```

Syntax: clear lldp statistics [**ports ethernet** *slot num/portnum* | **all**]

If you do not specify any ports or use the keyword **all**, by default, the system will clear lldp statistics on all ports.

You can list all of the ports individually, use the keyword **to** to specify ranges of ports, or a combination of both. To apply the configuration to all ports on the device, use the keyword **all** instead of listing the ports individually.

Brocade NetIron XMR Series and Brocade NetIron MLX Series Link Aggregation

NOTE
This chapter is applicable only to the Brocade NetIron XMR Series and Brocade NetIron MLX Series devices.

This chapter describes how to configure Link Aggregation Groups (LAG) for the Brocade NetIron XMR Series and Brocade NetIron MLX Series. You can use a single interface to configure any of the following LAG types:

- **Static LAGs** - These LAG groups are manually-configured aggregate links containing multiple ports.
- **Dynamic LAGs** - This LAG type uses the Link Aggregation Control Protocol (LACP), to maintain aggregate links over multiple port. LACP PDUs are exchanged between ports on each device to determine if the connection is still active. The LAG then shuts down ports whose connection is no longer active.
- **Keep Alive LAGs** - In a Keep Alive LAG a single connection between a single port on 2 Brocade devices is established. In a keep alive LAG, LACP PDUs are exchanged between the 2 ports to determine if the connection between the devices is still active. If it is determined that the connection is no longer active, the ports are blocked.

NOTE
The new LAG configuration procedures supersede the previous configurations procedures for LAGs and Dynamic Link Aggregation.

LAG formation rules

The LAG formation rules are mentioned below:

- The 10Gx24-DM module ports can only be part of LAGs exclusively consisting of 24x10G ports. A LAG cannot have a mix of 24x10G module ports and any other 10G module ports.
- A port can only be a member of one LAG, and that LAG must be static, dynamic or a keep-alive LAG.
- The maximum number of port members that may be assigned to a LAG is dependent on the number of trunks specified by the *system-max trunk-num* value.
- The system supports up to 64 port IDs for Brocade NetIron CES and Brocade NetIron MLX devices when **snmp-server max-ifindex-per-module 64** is configured.
- All ports configured in a LAG must be of equal bandwidth. For example all 10 G ports.
- All ports configured in a LAG must be configured with the same port attributes.
- LAG formation rules are checked when a static or dynamic LAG is deployed.
- A LAG must have its primary port selected before it can be deployed.
- All ports configured in a LAG must be configured in the same VLAN.
- All ports must have the same PBR configuration before deployment. During deployment, the configuration on the primary port is replicated to all ports. On undeployment, each port inherits the same PBR configuration.

- All static LAG ports must have the same LACP BPDU forwarding configuration.
- A LAG member and an individual port cannot use the same name.
- VLAN and inner-VLAN translation

The LAG is rejected if any LAG port has VLAN or inner-VLAN translation configured

- Layer 2 requirements:

The LAG is rejected if the LAG ports:

- – Do not have the same untagged VLAN component.
  – Do not share the same SuperSpan customer ID (CID).
  – Do not share the same VLAN membership or do not share the same uplink VLAN membership
  – Do not share the same protocol-VLAN configuration
  – Are configured as mainly primary and secondary interfaces
  – Static LAG deployment will fail if the if LACP BPDU forwarding is disabled on the primary port and enabled on one or more of the secondary ports.
- Layer 3 requirements:

The LAG is rejected if any of the secondary LAG port has any Layer 3 configurations, such as IPv4 or IPv6 address, OSPF, RIP, RIPNG, IS-IS, and so on.

- Layer 4 (ACL) requirements:
  – All LAG ports must have the same ACL configurations; otherwise, the LAG is rejected.
  – A LAG cannot be deployed if any of the member ports has ACL-based mirroring configured on it.
  – A port with ACL-based mirroring configured on it cannot be added to a LAG.
- The router can support up to 256 LAGs, and each LAG can contain up to 64 member ports.
  – If the router is configured to support 32 LAGs by using the **system-max trunk-num** command, the maximum number of LAG ports is 64.
  – If the router is configured to support 64 LAGs by using the **system-max trunk-num** command, the maximum number of LAG ports is 32.
  – If the *system-max trunk-num* value is set to 256, the maximum number of LAG ports supported is 8.
  – The default system-max trunk-num value is set to 128, and each LAG can have up to 16 member ports
  – For 40G and 100G ports, the number of LAG FIDs is 128. The configurable ranges are from 2 to 64 LAGs.
- When configuring a static or dynamic LAG, if trunk load sharing type is set to "per-packet" the maximum number of "per-packet" trunks is set to 4.
- Ports can be in only one LAG group. All the ports in a LAG group must be connected to the same device at the other end. For example, if port 1/4 and 1/5 in Device 1 are in the same LAG group, both ports must be connected to ports in Device 2 or in Device 3. You cannot have one port connected to Device 2 and another port connected to Device 3.
- All LAG member properties must match the primary port of the LAG with respect to the following parameters:
  – Port tag type (untagged or tagged port)
  – Port speed and duplex
  – TOS-based Configuration - All ports in the LAG must have the same TOS-based QoS configuration before LAG deployment, During deployment the configuration on the primary port is replicated to all ports and on undeployment, each port inherits the same TOS-based QoS configuration.

To change port parameters, you must change them on the primary port. The software automatically applies the changes to the other ports in the LAG.

- Using the **system-max trunk-num** command, the device can support the following LAG/member port configurations:
  – 256 LAGs with each containing 8 member ports.

  - 128 LAGs with each containing 16 member ports.
  - 64 LAGs with each containing 32 member ports.
  - 32 LAGs with each containing 64 member ports.
- Using the **system-max trunk-num-100g** command, the device can support the following 40 GbE and 100 GbE LAG scalability configurations.
  - 64 LAGs with each containing 2 member ports.
  - 32 LAGs with each containing 4 member ports.
  - 16 LAGs with each containing 8 member ports.
  - 8 LAGs with each containing 16 member ports.
  - 4 LAGs with each containing 32 member ports.
  - 2 LAGs with each containing 64 member ports.
- The total number of ports in a trunk is controlled by the **system-max trunk-num** command for both non-100G and 100G trunks.
- Make sure the device on the other end of the LAG link can support the same number of ports in the link.

Mixed port LAG support for 2x100GbE module:

LAGs can be formed between ports that have the same speed. For example, the default speed for 10G or 1G port is auto for 4x10GbE card, similar to 2x100GbE card. LAG is supported between 10G port and 1G port, if the port speed is in AUTO configuration.

- If a 2x100GbE port is added to a LAG, then the port should be configured to operate in AUTO mode.
- The 2x100GbE port supports LAG formation with any other 10G or 1G port.
- If 2x100GbE port is speed configured and is added to LAG, then the LAG deployment fails with error message.
- If a 2x100GbE port is added to a LAG and if the ports of a LAG are of different operating speed, packet loss is expected.
- For 2x100GbE ports "confirm-port-up" value check is not performed at LAG deployment.

TABLE 20 The LAG/member configuration

Maximum number of 1/10G LAGs	Maximum number of 1/10G LAG ports	Maximum number of 100/40G LAG ports	Maximum number of 100/40G LAGs possible	Maximum number of LAGs in the system
256	8	2 to 8	16 for 8-ports 64 for 2-ports	256
128	16	2 to 16	8 for 16-ports 64 for 2-ports	128
64	32	2 to 32	4 for 32-ports 64 for 2-ports	64
32	64	4 to 64	2 for 64-ports 32 for 4-ports	32

Figure 3 displays an example of a valid, Keep ALIVE LAG link between two devices. This configuration does not aggregate ports but uses the LACP PDUs to maintain the connection status between the two ports.

FIGURE 3 Example of a 1-port keep alive LAG

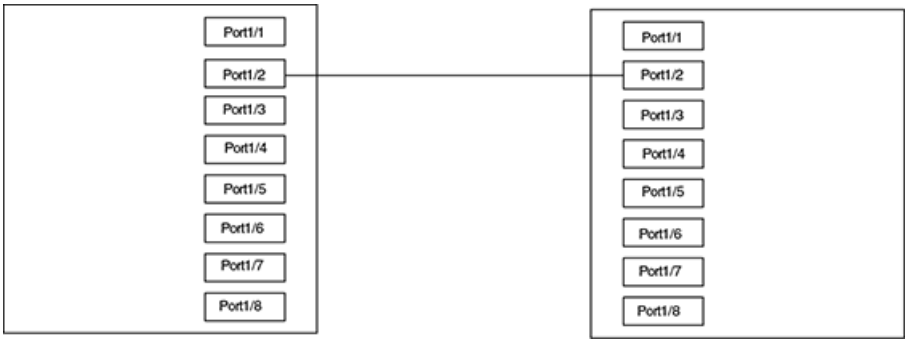


Figure 4 shows an example of a valid 2-port LAG link between devices where the ports on each end are on the same interface module. Ports in a valid 2-port LAG on one device are connected to two ports in a valid 2-port LAG on another device.

FIGURE 4 Example of 2-port LAG

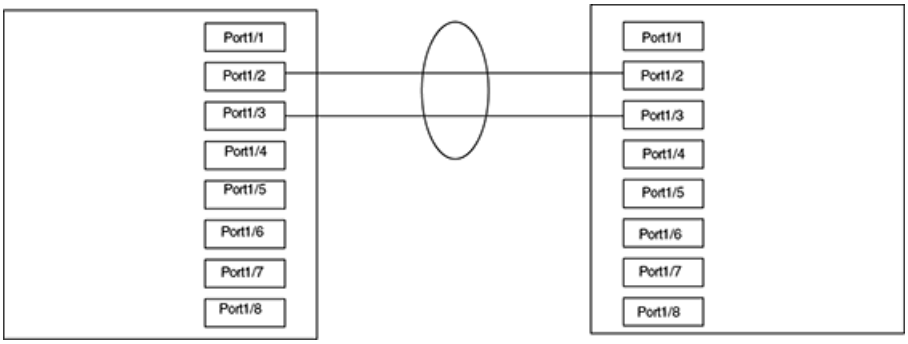
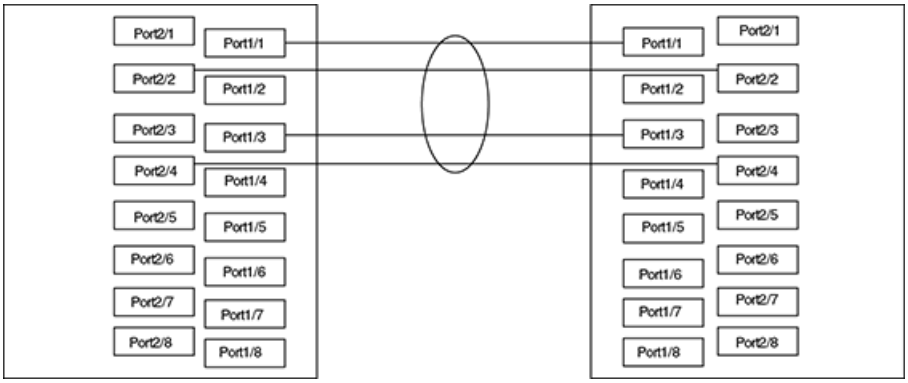


Figure 5 shows an example of two devices connected over a 4 port LAG where the ports on each end of the LAG are on different interface modules.

FIGURE 5 Examples of multi-slot, multi-port LAG



LAG load sharing

Brocade devices can be configured for load sharing over a LAG by either of the following methods:

- Hash Based Load Sharing
- Per Packet Load Sharing

Each of these methods, that are described in the following sections, are configured per LAG using the **trunk-type** command as described in Configuring load sharing type on page 131.

Hash based load sharing

The Brocade device shares the traffic load evenly across the ports in LAG group, while ensuring that packets in the flow are not reordered. Individual flows are assigned a LAG index to identify them. An improved hash based load sharing algorithm has the following enhancements:

- Better Distribution
- Support for 32-port LAGs
- An increased number of fields in the packet header that can be used for load balancing
- Enhanced load sharing in configurations of ECMP with LAGs.

Traffic from each flow is then distributed across the ports in the LAG group using a hash index as follows:

- For Layer 2 switching, the hash index is based on the following:
  - IPv4 or IPv6 traffic: source MAC address and destination MAC address, IPv4v6 source and destination address, VLAN-ID, IPv4 protocol number or IPv6 next header and TCP or UDP source port and TCP or UDP destination port.

    NOTE
    TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or the **load-balance force-l4-hashing** command is configured.

- Layer-2 packets with an MPLS payload: source MAC address and destination MAC address, VLAN ID, Inner VLAN ID (for double-tagged packets), Ethertype, and up to 3 MPLS Labels.

  NOTE
  For double-tagged packets, Ethertype is not used and the TPID of the inner TAG must be 0x8100 to be considered a double-tagged packet.

- GRE encapsulated IPv4 traffic: source MAC address and destination MAC address, IPv4 source and destination address, IPv4 protocol number. VLAN-ID, and TCP or UDP source port and TCP or UDP destination port. Also, inner IPv4 source address and inner destination IPv4 address are used if there is at least one GRE or IPv6 tunnel configured.

  NOTE
  TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured.

- IPv6 tunnel encapsulated IPv6 traffic: source MAC address and destination MAC address, IPv6 source and destination address, TCP or UDP source port and TCP or UDP destination port, IPv6 next header, and VLAN ID.

  NOTE
  TCP or UDP Destination and Source port is used under the following conditions:- Packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured. For 6over4 encapsulated packets, inner IPv6 destination address and inner IPv6 source address are used if there is at least one GRE or IPv6 tunnel configured.

- Layer-2, non-IPv4, IPv6 or non-MPLS packets: source MAC address, destination MAC address, VLAN ID, and Ether type.
- For Layer 3-Routing, the hash index is based on the following:
  - IPv4 or IPv6 traffic: source MAC address and destination MAC address, IPv4v6 source and destination address, VLAN-ID, IPv4 protocol number or IPv6 next header and TCP or UDP source port and TCP or UDP destination port.

  NOTE
  TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured.

- GRE encapsulated IPv4 traffic: source MAC address and destination MAC address, IPv4 source and destination address, IPv4 protocol number. VLAN-ID, and TCP or UDP source port and TCP or UDP destination port. Also, inner IPv4 source address and inner destination IPv4 address are used if there is at least one GRE or IPv6 tunnel configured.

  NOTE
  TCP or UDP Destination and Source port is used under the following conditions:- Packet is non-fragmented and without option, and packet is a TCP or UDP packet- or, the **load-balance force-l4-hashing** command is configured.

- IPv6 tunnel encapsulated IPv6 traffic: source MAC address and destination MAC address, IPv6 source and destination address, TCP or UDP source port and TCP or UDP destination port, and IPv6 next header. and VLAN-ID.

  NOTE
  TCP or UDP Destination and Source port is used under the following conditions:- Packet is a TCP or UDP packet, or, the **load-balance force-l4-hashing** command is configured. For 6over4 encapsulated packets, inner IPv6 destination address and inner IPv6 source address are used if there is at least one GRE or IPv6 tunnel configured.

- For MPLS switching, the hash index is based on the following:
  - L2VPN traffic: outer source MAC address and outer destination MAC address, up to two MPLS Labels, VLAN ID, inner source MAC address and inner destination MAC address, If packet payload is an IPv4 or v6 packet: IPv4v6 source and destination address, IPv4 Protocol Number or IPv6 Next Header ID of the payload are used.
  - L3VPN traffic or IP shortcut traffic: outer source MAC address and outer destination MAC address, VLAN ID, inner source IPv4v6 address and inner destination IPv4v6 address, IPv4 Protocol Number or IPv6 Next Header ID, TCP source port and TCP destination port, UDP source port and UDP destination port, and up to two MPLS Labels.
  - MPLS packets with 3 labels: outer source MAC address and outer destination MAC address, VLAN ID, and all 3 MPLS Labels.

NOTE
For transit LSRs please note the following: The **load-balance speculate-mpls-ip** command must be active. It is on by default. If the **load-balance speculate-mpls-ip** command has been configured to be inactive, and the **load-balance speculate-mpls-enet** command is active, the packet will be processed like an L2VPN packet. If both commands are configured to be inactive, no inner layer 2 or layer 3 headers are considered but up to 3 MPLS labels are used for hashing.

Options for hash based load sharing

The following options can be used to refine the hash calculations used for LAGs:

- Speculate UDP or TCP Headers
- Mask Layer-4 Source and Destination Port Information
- Hash Diversification

Each of these options when configured apply to both IP Load Sharing and LAG Load sharing. They are described in detail in Configuring IP Chapter.

Load sharing for MPLS LAGs

Load sharing on MPLS LAG involves traffic flows that include the MPLS Inner and Outer Labels. These can be used exclusively or in combination with the IP and MAC source and destination addresses to determine the LAG index for a traffic flow.

Using IP source and destination addresses for load sharing

You can use the **load-balance speculate-mpls-ip** command to include the IP source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
device(config)# load-balance speculate-mpls-ip all
```

Syntax: [no] load-balance speculate-mpls-ip [**all** | **slot-number** | **slot-number np-id**]

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The **load-balance speculate-mpls-ip** command will hash only on the IP portion.

Using MAC source and destination addresses for load sharing

You can use the **load-balance speculate-mpls-enet** command to include the MAC source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
device(config)# load-balance speculate-mpls-enet all
```

Syntax: [no] load-balance speculate-mpls-enet [**all** | **slot-number** | **slot-number np-id**]

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE
The **load-balance speculate-mpls--enet** command will hash only on the Ethernet header portion.

Per packet server LAG load sharing

Per packet LAG load balancing is a type of LAG that load balances traffic on a per-packet basis, as compared to traditional server LAG load-balancing which balances traffic based on packet content such as source or destination addresses. In per packet server LAG load balancing, the packet processor (PPCR) on each module selects a port in the per packet server LAG to forward traffic in a round-robin fashion. For example, if the first port of the per packet server LAG is currently selected, the second port of the per-packet server LAG will be used next, and so on. Consequently, traffic is evenly distributed among all of the ports that are configured in a per packet server LAG.

Traffic that can be forwarded out of a per-packet LAG includes Layer 2 switching traffic, Layer 3 routing traffic, L3VPN (2547) traffic, VLL and VPLS traffic.

Configuring a LAG

The following configuration procedures are used to configure a LAG. Depending upon whether you are configuring a static, dynamic or keep-alive LAG, the configuration procedures may or may not apply as described:

- **Creating a Link Aggregation Group** - Required for all static, dynamic or keep alive LAGs.
- **Adding Ports to a LAG** - Required for all static, dynamic, or keep alive LAGs. A keep alive LAG contains only one port while static and dynamic LAGs can have 2 to 32 ports.
- **Configuring the Primary Port for a LAG** - Required for all static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Configuring the Load Sharing Type** - Optional for all static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Specifying the LAG Threshold for a LAG Group** - Optional for static and dynamic LAGs. Since a keep alive LAG contains only one port, it is unnecessary to configure this parameter.
- **Configuring LACP Port Priority** - Optional for dynamic and keep alive LAGs.
- **Configuring an LACP Timeout** - Optional for dynamic and keep alive LAGs.
- **Configuring LACP BPDU Forwarding** - Optional for static LAGS only since LACP BDUs are discarded (dropped) on ports in which a static LAG has been configured as the default setting.

Creating a LAG using the LAG ID option

Before setting-up ports or configuring any other aspects of a Link Aggregation Group (LAG), you must create it first.

You can either assign a LAG ID explicitly or it will be automatically generated by the system. The LAG ID stays the same across system reload and hitless upgrade.

The command to configure LAGs allows explicit configuration of the LAG ID for static and dynamic LAGs.

To create a LAG with the LAG ID option, enter a command such as the following.

```
device(config)# lag blue static
device(config-lag-blue)#
```

Syntax: [no] lag name [**static** | **dynamic**] [**id number**]

The **ID** parameter is optional. The value of the **ID** parameter that you can enter is from 1 to 256. If you do not enter a LAG **ID** , the system will generate one automatically. Once the LAG **ID** is generated the system will save it in the configuration file along with the LAG name, therefore the value will stay the same across system reload.

> NOTE
> The LAG **ID** parameter is for static and dynamic LAGs only. No explicit configuration of a LAG **ID** is allowed on keepalive LAGs.

The **static** parameter specifies that the LAG with the name specified by the *lag-name* variable will be configured as a static LAG.

The **dynamic** option specifies that the LAG with the name specified by the *lag-name* variable will be configured as a dynamic LAG.

Configuration considerations

LAG IDs are unique for each LAG in the system. The same LAG IDs cannot be assigned to two or more different LAGs. If a LAG ID is already used, the CLI will reject the new LAG configuration and display an error message that suggests the next available LAG ID that can be used.

```
device(config)#lag lag3 static id 123
Error: LAG id 123 is already used. The next available LAG id is 2
.
```

LAG configured with LAG ID 124.

```
!
lag "lag1" static id 124
 ports ethernet 1/2 to 1/3
 primary-port 1/3
 deploy
!
```

The **show lag** command and the output.

```
device(config)# show lag
Total number of LAGs:          1
Total number of deployed LAGs: 1
Total number of s created:1 (127 available)
LACP System Priority / ID:     1 / 0000.0001.c000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "lag1" ID 124 (static Deployed) ===
LAG Configuration:
   Ports:         ethe 1/2 to 1/3
   Port Count:    2
   Primary Port:  1/3
    Type:    hash-based
Deployment:   ID 124, Active Primary 1/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC Name
1/2   Up   Forward   Full 10G   124   No  level0 0000.0001.c002
1/3   Up   Forward   Full 10G   124   No  level0 0000.0001.c002
```

Creating a keepalive LAG

To create a **keep-alive** LAG, enter the following.

```
device(config)# lag lag1 keep-alive
```

Syntax: [no] lag name keep-alive

The **keep-alive** option specifies that the LAG with the name specified by the *lag-name* variable will be configured a keep-alive LAG. The keep-alive LAG option allows you to configure a LAG for use in keep alive applications similar to the UDLD feature.

Disabling the Detection of Remote LACP Configuration Removal

The **lacp-cfg-det-dis** command is a global command and is used to disable detecting remote end LACP configuration removal. By default, this feature this is enabled. To disable this feature, enter a command such as the following:

```
device(config)# lacp-cfg-det-dis
```

Syntax: lacp-cfg-det-dis

> NOTE
> When you have interoperability with another vendor's device, you will need to disable this feature as other vendors devices will remove the fiber.

Modifying an existing LAG name

The Brocade NetIron devices support changing an existing LAG name without deleting and recreating the LAG.

Use the **update-lag-name** command to modify an existing LAG name. This command works for all LAG types, such as static, dynamic, and keepalive LAGs.

```
device(config)# lag blue
device(config-lag-blue)# update-lag-name brocade
```

Syntax: update-lag-name *new-name*

> NOTE
> The modified LAG name should be unique across all the LAG names that are available.

Adding Ports to a LAG or Deleting Ports from a LAG

A static or dynamic LAG can consist of from 2 to 32 ports of the same type and speed that are on any interface module within the Brocade chassis. A keep alive LAG consists of only one port.

To configure the static LAG named "blue" with two ports, use the following command:

```
device(config)# lag blue static
device(config-lag-blue)# ports ethernet 3/1 ethernet 7/2
```

Syntax: [no] ports ethernet *slot/port* [**to** *slot/port*] [**ethernet** *slot/port*]

The ports added to a LAG can be of type **ethernet** as specified for the **slot/port** where they reside. The ports can be added to the LAG sequentially as shown in the following example:

```
device(config-lag-blue)# ports ethernet 3/1 ethernet 7/2 ethernet 4/3 ethernet 3/4
```

A range of ports from a single interface module can be specified. In the following example, Ethernet ports 1, 2, 3 and 4 on the interface module in slot 3 are configured in a single LAG:

```
device(config-lag-blue)# ports ethernet 3/1 to 3/4
```

Additionally, you can mix a range of ports from one interface module with individual ports from other interface modules to form a LAG as shown in the following:

```
device(config-lag-blue)# ports ethernet 3/1 to 3/4 ethernet 10/2
```

Using the **no** option allows you to remove ports from a LAG. For example, you can remove port 3/4 from the LAG created above, as shown in the following:

```
device(config-lag-blue)# no ports ethernet 3/4
```

Ports can be added to an undeployed LAG or to currently deployed LAG using the commands described. For special considerations when adding ports to or deleting ports from a currently deployed LAG, refer to the following sections:

Adding a port to a currently deployed LAG on page 136

Deleting a port from a currently deployed LAG on page 136

Configuring the primary port for a LAG

The primary port must be explicitly assigned using the **primary-port** command.

To designate the primary port for the static LAG "blue", use the following command.

```
device(config)# lag blue static
device(config-lag-blue)# primary-port 3/2
```

Syntax: [no] primary-port slot/port

Once a primary port has been configured for a LAG, all configurations that apply to the primary port are applied to the other ports in the LAG.

> NOTE
> This configuration is only applicable for configuration of a static or dynamic LAG.

Configuring load sharing type

Individual LAGs can be configured to perform load sharing over the ports in the LAG using either a hash based or per packet method, as shown in the following .

```
device(config)# lag blue static
device(config-lag-blue)# trunk-type hash-based
```

Syntax: [no] trunk-type hash-based | per-packet

> NOTE
> This configuration is only applicable for configuration of a static or dynamic LAGs.

Specifying the LAG threshold

Trunk threshold brings the LAG down if the trunk threshold condition is not met by the LAG.

Syntax: [no] trunk-threshold *number*

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

> NOTE
> This configuration is only applicable for configuration of a static or dynamic LAG.

Below are the different behavioral patterns for static and dynamic LAG.

Static LAG behavior

You can configure the Brocade device to disable all of the ports in a LAG when the number of active member ports drops below a specified threshold value. For example, if a LAG has 8 ports, and the threshold for the LAG is 5, then the LAG is disabled if the number of available ports in the LAG drops below 5. If the LAG is disabled, then traffic is forwarded over a different link or LAG.

For example, the following commands establish a LAG consisting of four ports, and then establish a threshold of three ports for this LAG.

```
device(config)# lag blue static
device(config-lag-blue)# ports ethernet 3/1 to 3/4
device(config-lag-blue)# trunk-threshold 3
```

In this example, if the number of active ports drops below three, then all the ports are disabled in the LAG.

When a LAG is down because of not meeting the LAG condition, the LAG is kept intact and it is re-enabled if enough ports become active to reach the threshold.

> NOTE
> The **trunk-threshold** command should be configured only at one end of the trunk. If it is set on both sides, link failures result in race conditions and change in functionality.

Dynamic LAG behavior

Unlike in static LAGs, if the number of active ports in a dynamic LAG falls below the threshold value, the ports are logically blocked and out-of-sync is signaled on all the active member ports in that LAG until it satisfies the trunk threshold condition again.

For example, the following commands establish a LAG consisting of four ports, and then sets a threshold of three ports.

```
device(config)# lag red dynamic
device(config-lag-blue)# ports ethernet 3/1 to 3/4
device(config-lag-blue)# trunk-threshold 3
```

In this example, if the number of active ports drops below 3, then all the ports in the LAG are logically blocked. An out-of-sync is signaled on all the active member ports in that LAG until it satisfies the trunk threshold condition again.

> NOTE
> Configure the same LACP Trunk threshold value on both sides of the LAG for better performance or for connecting to the third-party devices.

> NOTE
> Configuring the LACP trunk threshold on a third-party device may occasionally cause performance or incorrect traffic flow issues. To avoid this issue, configure the trunk threshold on either the Brocade NetIron MLX Series device only or on both the Brocade NetIron MLX Series device and the third-party device.

Configuring an LACP port priority

In a dynamic or keep-alive LAG, a port priority can be configured at the global level.

```
device(config)# lag blue dynamic
device(config-lag-blue)# lacp-port-priority 100000
```

Syntax: [no] lacp-port-priority slot/port *number*

> NOTE
> This configuration is only applicable for configuration of a dynamic or keep-alive LAGs.

Configuring an LACP system priority

In a dynamic or keep-alive LAG, a system priority can be configured at global level.

```
device(config)# lacp system-priority 4
```

Syntax: [no] lacp system-priority *number*

The number value specifies the value of the LACP system priority. This can be a value from 1 to 65535. The default system-priority value is 1.

NOTE
This configuration is only applicable for configuration of a dynamic or keep-alive LAGs.

NOTE
In a system configuration with multiple MCT peers, the LACP system priority on both the MCT nodes should be same.

Configuring an LACP timeout

In a dynamic or keep-alive LAG, a port's timeout can be configured as short (3 seconds) or long (90 seconds). After you configure a port timeout, the port remains in that timeout mode whether it is up or down and whether or not it is part of a LAG.

All the ports in a LAG should have the same timeout mode. This requirement is checked when the LAG is enabled on the ports. For example, to configure a port for a short LACP timeout, use the following command.

```
device(config)# lag blue dynamic
device(config-lag-blue)# lacp-timeout short
```

Syntax: [no] lacp-timeout [**long** | **short**]

To delete the configuration, use the **no** form of this command.

The **long** keyword configures the port for the long timeout mode-90 seconds. With the long timeout, an LACPDU is sent every 30 seconds. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transition to the appropriate state based on its current state.

The **short** keyword configures the port for the short timeout mode--3 seconds. In the short timeout configuration, an LACPDU is sent every second. If no response comes from its partner after 3 LACPDUs are sent, a timeout event occurs, and the LACP state machine transitions to the appropriate state based on its current state.

If you specify neither **long** nor **short**, the state machine operates based on the standard IEEE specification as its default behavior. The original IEEE specification says that the state machine starts with short the timeout and moves to the long timeout after the LAG is established. However, sometimes a vendor's implementation always uses either the short timeout or the long timeout without changing the timeout. Brocade provides this command so that you can configure Brocade devices to interoperate with other vendor's devices.

NOTE
This configuration is applicable to the configuration of dynamic or keep-alive LAGs only.

Configuring LACP BPDU Forwarding

Enabling and Disabling LACP BPDU Forwarding on a Port

For scenarios in which static LAG ports require LACP BPDU packet forwarding, you can issue the **forward-lacp** command in the interface configuration mode. Once LACP Forwarding has been enabled on a static LAG, all the LACP BPDUs will follow regular packet forwarding actions.

When LACP forwarding is enabled, the link OAM packets received on the LACP forwarding enabled interface will be processed and flooded on the VLAN. If the LACP forwarding is not enabled, the link OAM packets will be processed and then dropped.

To enable LACP BPDU forwarding, enter the LACP Forwarding command as follows.

```
device(config-if-e1000-3/5)# forward-lacp
```

Syntax: forward-lacp

To disable LACP BPDU forwarding, enter the lacp-forwarding command as follows. When a static LAG is undeployed the LACP BPDU forwarding state of the LAG will be retained on the individual ports.

```
device(config-if-e1000-3/5)# [no]
forward-lacp
```

Syntax: [no] forward-lacp

The **forward-lacp** option specifies that the port of the specified static LAG will be configured for LACP-BPDU forwarding. If the specified port is a dynamic or keep alive LAG, an error message will be displayed.

Enabling and Disabling LACP BPDU Forwarding on a LAG

> NOTE
> The forward-lacp command must be issued on the physical port configuration, not in LAG configuration.

When the LACP forwarding is enabled on the primary port of the static LAG, the LACP BPDU forwarding is enabled on all ports of the LAG when the LAG is deployed. When the static LAG is undeployed the BPDU forwarding state is retained.

> NOTE
> LACP BPDU forwarding is not supported for any port of dynamic or keep alive LAGs.

- If LACP BPDU forwarding is enabled on the primary and secondary ports, the static LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is enabled on the primary port and disabled on the secondary ports, the static LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is disabled on the primary and secondary ports, the static LAG deployment will be successful and LACP BPDU forwarding will be disabled on the LAG ports.

Deploying a LAG

After configuring a LAG, you must explicitly enable it before it begins aggregating traffic. This task is accomplished by executing the **deploy** command within the LAG configuration. After the **deploy** command runs, the LAG is in the aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG. The running configuration will no longer display deployed LAG ports other than the primary port.

To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keep-alive LAGs. After a non keep-alive LAG is deployed, a LAG is formed. If there is only one port in the LAG, a single port LAG is formed. For a dynamic LAG, LACP is started for each LAG port. For a keep-alive LAG, no LAG is formed and LACP is started on the LAG port.

You can deploy a LAG as shown in the following for the "blue" LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
```

Syntax: [no] deploy [**forced** | **passive**]

When the **deploy** command is executed:

For a static and dynamic LAGs, the current LAG veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a LAG is formed with all the ports in the LAG.

For dynamic LAGs, by default all LAG ports will be on the **active** mode and LACP is activated on all LAG ports. If you specify **passive** mode, the LACP ports do not initiate the aggregation aggressively and ports will respond to LACP packets only when it receives LACP PDUs.

For a keep-alive LAGs, no LAG is formed, and LACP is started on the LAG port.

Once the **deploy** command is issued, all LAG ports will behave like a single port.

If the **no deploy** command is executed, the LAG is removed. For dynamic LAGs, LACP is de-activated on all of the LAG ports.

If the **no deploy** command is issued and more than 1 LAG port is not disabled the command is aborted and the following error message is displayed: "Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted." Using the **forced** keyword with the **no deploy** command in the previous situation, the un-deployment of the LAG is executed.

Commands available under LAG once it is deployed

Once a LAG has been deployed, the following configurations can be performed on the deployed LAG:

- Configuring ACL-based Mirroring
- Disabling Ports within a LAG
- Enabling Ports within a LAG
- Monitoring and Individual LAG Port
- Assigning a name to a port within a LAG
- Enabling sFlow Forwarding on a port within a LAG
- Setting the sFlow Sampling Rate for a port within a LAG
- Configuring LACP BPDU Forwarding

Configuring ACL-based mirroring

To configure ACL-based mirroring for all ports in a LAG, configure it on the primary port of the LAG at the interface configuration level (see Configuring IP Chapter). ACL-based mirroring can be configured for an individual member port within a LAG by using the **acl-mirror-port** command, as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# acl-mirror-port ethe-port-monitored 3/1 ethernet 3/2
```

In this example, traffic on Ethernet port 3/1 (a member port of LAG "blue") will be mirrored to Ethernet port 3/2.

Syntax: [no] acl-mirror-port { ethe-port-monitored slot/port | named-port-monitored name } ethernet slot/port

Use the **ethe-port-monitored** option with the appropriate *slot/port* variable to specify an Ethernet port for which you want to provide ACL mirroring.

Use the **named-port-monitored** option with the appropriate *slot/port* variable to specify a named port for which you want to provide ACL mirroring.

The **ethernet** keyword precedes the *slot/port* variable identifying the port which will receive the mirrored packets.

> NOTE
> A port with ACL-based mirroring already configured on it cannot be added to a LAG, and a LAG cannot be deployed if any of its member ports has ACL-based mirroring. To use ACL-based mirroring on a LAG member port, deploy the LAG, then configure mirroring on the member port. If a port is removed from a LAG, ACL-based mirroring will be removed from that port, and if a LAG is deleted mirroring will be removed from all member ports.

Disabling ports within a LAG

You can disable an individual port within a LAG using the disable command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# disable ethernet 3/1
```

Syntax: [no] disable ethernet [**slot/port**] | **named** [**name**]

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to disable.

Use the **named** option with the appropriate *[slot/port]* variable to specify a named port within the LAG that you want to disable.

Enabling ports within a LAG

You can enable an individual port within a LAG using the enable command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# enable ethernet 3/1
```

Syntax: [no] enable ethernet [**slot/port**] | **named** [**name**]

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to enable.

Use the **named** option with the appropriate *[slot/port]* variable to specify a named port within the LAG that you want to enable.

Adding a port to a currently deployed LAG

Ports can be added to a currently deployed LAG. Adding a port to a deployed LAG uses the same procedures as described in Adding Ports to a LAG or Deleting Ports from a LAG on page 130. When you add ports to a deployed LAG, the MAC address of the port being added is changed to that of the primary port of the LAG to which it is being added.

When adding a port to a currently deployed static LAG the LACP BPDU forwarding configuration must be the same as the LAG. Follow the procedure on Enabling and Disabling LACP BPDU Forwarding on a Port on page 133.

Deleting a port from a currently deployed LAG

Ports can be deleted from a currently deployed LAG. Deleting a port in a currently deployed LAG uses the same procedures as described in Adding Ports to a LAG or Deleting Ports from a LAG on page 130. However, when deleting ports from a currently deployed LAG you must consider the following:

- The primary port cannot be removed.
- If removal of a port will result in the trunk threshold value becoming greater than the number of ports in the LAG, the port deletion will be rejected.
- The port being deleted must be in the "disabled" state or you must use the forced option (as described in the following command syntax) when deleting it from a currently deployed LAG. Otherwise, the deletion request will be denied and the following error message will be displayed: "Error: ports to be deleted from the deployed LAG are not disabled, deleting these ports from the LAG may form a loop - aborted."
- When a port is deleted from a deployed static LAG, the LACP BDPU forwarding state of the LAG will be retained on the deleted port.

To delete port 3/1 which is in the "enabled" state from a currently deployed LAG named "blue", use the following command:

```
device(config)# lag blue static
device(config-lag-blue)# no ports ethernet 3/1 forced
```

Syntax: no ports ethernet slot/port [**to slot/port**] [**ethernet slot/port**] [**forced**]

This command operates as described in Adding Ports to a LAG or Deleting Ports from a LAG on page 130 except for the **forced** option which is described in the following:

The **forced** option to the **no ports** command deletes a port from a currently deployed LAG even if it is currently in the "enabled state". Because deleting an enabled port from a currently deployed LAG can cause a loop to be formed, we recommend that you disable any port being removed from a LAG before removing it. Only use the **forced** option when you are confident that a loop will not be created in your network topology.

> NOTE
> When a port is deleted from a currently deployed LAG, the MAC address of the port is changed back to its original value.

Monitoring an individual LAG port

By default, when you monitor the primary port in a LAG group, aggregated traffic for all the ports in the LAG is copied to the mirror port. You can configure the device to monitor individual ports in a LAG including Ethernet, or named ports. You can monitor the primary port or another member port individually.

> NOTE
> You can use only one mirror port for each monitored LAG port. To monitor traffic on an individual port in a LAG group, enter commands such as the following.

This command enables monitoring of an individual port within a LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# monitor ethe-port-monitored 3/1 ethernet 10/3 both
```

Syntax: [no] monitor ethe-port-monitored [**slot/port**] | **named-port-monitored** [**name**] | **ethernet** [**slot/port**] [**input** | **output** | **both**]

Use the **ethe-port-monitored** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to monitor.

Use the **named-port-monitored** option with the appropriate *[slot/port]* variable to specify a named port within the LAG that you want monitor.

The **ethernet** *slot/port* parameter specifies the port to which the traffic analyzer is attached.

The **input, output, and both** parameters specify the traffic direction to be monitored.

Assigning a name to a port within a LAG

You can assign a name to an individual port within a LAG using the **port-name** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# port-name orange ethernet 3/1
```

Syntax: [no] port-name text ethernet [**slot/port**]

The *text* variable specifies the port name. The name can be up to 50 characters long.

Use the **ethernet** option with the appropriate *[slot/port]* variable to apply the specified name to an Ethernet port within the LAG.

Refer to the *Brocade NetIron Administration Guide* for additional information on LAG naming conventions.

NOTE
The port name and LAG name cannot use the same name.

Enabling sFlow forwarding on a port in a LAG

You can enable sFlow forwarding on an individual port within a LAG using the **sflow-forwarding** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-forwarding ethernet 3/1
```

Syntax: [no] sflow-forwarding ethernet [**slot/port**] | **port-name** [**text**]

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify a Ethernet port within the LAG that you want to enable sFlow forwarding for.

Use the **port-name** option with the appropriate *[text]* variable to specify a named port within the LAG that you want to enable sFlow forwarding for.

Setting the sFlow sampling rate for a port in a LAG

NOTE
The NetIron CES and NetIron CER supports sflow sampling rate configuration per port basis. The Brocade MLXe, NetIron MLX, and NetIron XMR supports sflow sampling rate configuration per packet processer basis.

You can set the sFlow sampling rate for an individual port within a LAG using the **sflow-subsampling** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-subsampling ethernet 3/1 512
```

Syntax: [no] sflow-subsampling ethernet [**slot/port**] | **port-name** [**text**] **num**

Use the **ethernet** option with the appropriate *[slot/port]* variable to specify the Ethernet port within the LAG that you want to configure the sampling rate for.

Use the **port-name** option with the appropriate *[text]* variable to specify the named port within the LAG that you want to configure the sampling rate for.

The *num* variable specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. This can be a value between 512 - 1048576.

Configuring a dynamic LAG within a VRF

When configuring a dynamic LAG within a VRF, the following conditions must be considered:

- The dynamic LAG must be configured before adding it to a VRF.
- Before the LAG is deployed, all members must be in the default VRF.
- After the LAG is deployed, all LAG ports are in the LACP BLOCK state until the LACP protocol completes negotiation with the other end of the LAG.
- Once the LACP protocol negotiation is completed with the other end of the LAG, all the LAG ports are set to the FORWARD state.

- When a dynamic LAG within a VRF is undeployed, the primary port will stay in the VRF where the LAG was configured and the secondary ports of the LAG will return to the default VRF.

The following example uses the LAG and VRF commands to configure a LAG within a VRF.

```
device(config)# lag red dynamic
device(config-lag-red)# primary-port 3/2
device(config-lag-red)# ports ethernet 3/1 ethernet 7/2
device(config-lag-red)# exit
device(config)# interface ethernet 3/2
device(config-if-e10000-3/2)# vrf forwarding VPN1
device(config)# lag red dynamic
device(config-lag-red)# deploy
```

Configuring multicast dynamic load rebalancing on a LAG

In multicast, each forwarding (S,G) entry that has a Link Aggregation Group (LAG) port as an Outgoing Interface (OIF) is allocated only one of the member ports of the LAG for forwarding purposes. This member port is referred to as the forwarding port of the OIF of the (S,G) entry. A LAG is said to have balanced Multicast flows if all its member ports carry outgoing traffic for the same number of forwarding entries.

There are two ways of applying this feature, through a global configuration command that will force dynamic load rebalancing at all times, or via an exec level command that will trigger dynamic load rebalancing on demand.

Limitations

- The load balancing metric is determined by the number of forwarding (S,G) entries per LAG member port. Dynamic rebalancing attempts to balance this metric. This does not necessarily guarantee that multicast traffic bandwidth (bytes/sec) will be equally balanced across all member ports.
- When dynamic rebalancing is taking place, there will be dropped packets. This is because as the forwarding port for a LAG OIF changes, the FID and the MVID of the forwarding entry change as well, thus requiring a reprogramming of the PRAM before the changes take effect. This results in a brief disruption in traffic.
- Multicast dynamic load rebalancing on a LAG is only available on the Brocade NetIron XMR Series and Brocade NetIron MLX Series. This feature is not available in the Brocade NetIron CES Series and Brocade NetIron CER Series.

Configuring multicast dynamic load rebalancing

Once the dynamic load rebalancing has been configured, anytime a port in a LAG interface becomes active or a new port is added to the LAG interface, multicast traffic over the LAG interface will be rebalanced across all VRFs.

Dynamic load rebalancing has to be explicitly enabled (on all trunks and on all VRFs in the system) using the **ip multicast-routing lag rebalance** command.

```
device(config)#ip multicast-routing lag rebalance
```

Syntax: [no] ip multicast-routing lag rebalance

When enabling dynamic load balancing the specific IP version must be indicated. Use **ip** for IPv4 traffic and **ipv6** for IPv6 traffic.

Displaying LAG information

You can display LAG information for a Brocade device in either a **full** or **brief** mode.

The following example displays the **brief** option of the **show lag** command.

```
device# show lag brief
Total number of LAGs   : 2, 100g : 2
```

```
Total number of deployed LAGs  : 2, 100g : 2
Total number of trunks created : 2 (254 total available), 100g : 2 (14 total available)
LACP System Priority / ID      :1 / 0024.3883.3600
LACP Long timeout      :90, default: 90
LACP Short timeout     :3, default: 3
LAG               Type     Deploy Trunk Primary       Port List
100g_lag         static      Y    1     3/1        e 3/1
10g_lag          static      Y    2     2/1        e 2/1
1g_lag           static      Y    3     1/21       e 1/21
lag2            dynamic    Y    4     3/2       e 3/2
```

Syntax: show lag brief

Table 21 describes the information displayed by the **show lag brief** command.

The following example displays the full option of the **show lag** command.

```
device# show lag
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of s created:3 (125 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "d1" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 13/2 to 13/3 ethe 32/2
   Primary Port:  32/2
    Type:    hash-based
   LACP Key:      104
Deployment:   ID 3, Active Primary 3/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
3/2   Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
13/3  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
32/2  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
13/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
13/3       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
32/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
=== LAG "e" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 2/1 ethe 2/3 ethe 2/5
   Primary Port:  2/3
    Type:    hash-based
   LACP Key:      105
Deployment:   ID 1
Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
2/1   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/3   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/5   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
2/1        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/3        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/5        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
```

Syntax: show lag lag-name [ID] [name] [deployed] [dynamic] [Ethernet] [keep-alive] [static]

Using command this without options displays information for all LAGs configured on the device.

The *lag-name* variable allows you to limit the display to information for a specific LAG.

The **ID** option displays the output for the LAG specified by the ID.

The **name** displays the output for the LAG specified by the LAG name.

The **deployed** option limits the display to LAGs that are currently deployed.

The **dynamic** option limits the display to dynamic LAGs.

The **Ethernet** option displays the output for the specified Ethernet port.

The **keep-alive** option limits the display to keep alive LAGs.

The **deployed** option limits the display to static LAGs.

Optional commands include:

Syntax: show lag id

Syntax: show lag id *num_id*

Syntax: show lag ethernet *slot/port*

To display long port names, **set-lag-port-mode-wid** command. This command is useful if the ports names are long. In wide mode, the complete port name will be displayed and port type will not be displayed. In standard (non-wide) mode, only a portion of the port name is displayed if the port name is long and the port type is displayed. The following example shows the wide-mode display of a **show lag** command.

: Wide-Mode Display

```
device(config)#set-lag-port-mode-wid
device((config)#sh lag
Total number of LAGs:          2
Total number of deployed LAGs: 2
Total number of trunks created:2 (126 available)
LACP System Priority / ID:     1 / 00da.1111.2200
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "1234567890$%'-_@~`!(){}^#&abcdefghijklmnopqrstuvwXYZ" ID 4 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/3 to 31/10 e 31/15 to 31/20
   Port Count:    14
   Primary Port:  31/3
   Trunk Type:    hash-based
   LACP Key:      101
Port Individual Configuration:
   Port Name
   31/3 test2
Deployment:  Trunk ID 4, Active Primary none, base fid: 0x0810
Port  Link Port-State  Speed Tag MAC            Name
31/3  DisabNone        None  No  00da.1111.27a2 test2
31/4  DisabNone        None  No  00da.1111.27a2
31/5  DisabNone        None  No  00da.1111.27a2
31/6  DisabNone        None  No  00da.1111.27a2
31/7  DisabNone        None  No  00da.1111.27a2
31/8  DisabNone        None  No  00da.1111.27a2
31/9  DisabNone        None  No  00da.1111.27a2
31/10 DisabNone        None  No  00da.1111.27a2
31/15 DisabNone        None  No  00da.1111.27a2
31/16 DisabNone        None  No  00da.1111.27a2
31/17 DisabNone        None  No  00da.1111.27a2
31/18 DisabNone        None  No  00da.1111.27a2
31/19 DisabNone        None  No  00da.1111.27a2
31/20 DisabNone        None  No  00da.1111.27a2
=== LAG "NN" ID 6 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/11 to 31/14 e 31/21 to 31/24
   Port Count:    8
   Primary Port:  31/11
   Trunk Type:    hash-based
   LACP Key:      100
Port Individual Configuration:
   Port Name
   31/11test
Deployment:  Trunk ID 6, Active Primary 31/12, base fid: 0x0800
Port  Link Port-State  Speed Tag MAC            Name
31/11 Up   Forward     1G    No  00da.1111.27aa test
31/12 Up   Forward     1G    No  00da.1111.27aa
31/13 Up   Forward     1G    No  00da.1111.27aa
31/14 Up   Forward     1G    No  00da.1111.27aa
```

```
    31/21 Up    Forward     1G    No  00da.1111.27aa
    31/22 Up    Forward     1G    No  00da.1111.27aa
    31/23 Up    Forward     1G    No  00da.1111.27aa
    31/24 Up    Forward     1G    No  00da.1111.27aa
```

Syntax: [no] set-lag-port-mode-wid

Table 21 describes the information displayed by the **show lag** command.

TABLE 21 Show LAG information

This field...	Displays...
Total number of LAGS	The total number of LAGs that have been configured on the device.
Total number of Deployed LAGS	The total number of LAGs on the device that are currently deployed.
Total number of LAGs Created	The total number of LAGs that have been created on the LAG. The total number of LAGs available are shown also. Since keep-alive LAGs do not use a LAG ID, they are not listed and do not subtract for the number of LAGs available.
LACP System Priority /ID	The system priority configured for the device.The ID is the system priority which is the base MAC address of the device.
LACP Long timeout	The number of seconds used for the LACP Long timeout mode. This is only applicable for dynamic or keep-alive LAGs.
LACP Short timeout	The number of seconds used for the LACP Short timeout mode. This is only applicable for dynamic or keep-alive LAGs.
LACP BPDU Forwarding	Status of LACP BPDU forwarding on a static LAG: Disabled- LACP BDPU forwarding is disabled for all ports of the LAG, default setting. Enabled- LACP BDPU forwarding is enabled for all ports of the LAG.
The following information is displayed per-LAG in the show lag brief command.	
LAG	The name of the LAG.
Type	The configured type of the LAG: static, dynamic, or keep-alive
Deploy	Status of LAG deployment: Y - yes, LAG is deployed. N - no, LAG is not deployed.
LAG	The LAG ID number.
Primary	The primary port of the LAG.
Port List	The list of ports that are configured in the LAG.
The following information is displayed per-LAG the show lag command for each LAG configured.	
LAG Configuration	
Ports:	List of ports configured with the LAG.
Primary Port:	The primary port configured on the LAG.
LAG Type:	The load sharing method configured for the LAG: either hash-based or per-packet.
LACP Key	The link aggregation key for the LAG.
Deployment	
LAG ID	The LAG ID number.
Active Primary	The port within the LAG where most protocol packets are transmitted. This is not the same as the configured Primary Port of the LAG.
Port	The chassis slot and port number of the interface.
Link	The status of the link which can be one of the following:

TABLE 21 Show LAG information (continued)

This field...	Displays...
	• up • down
L2 State	The Layer 2 state for the port.
Dupl	The duplex state of the port, which can be one of the following: • Full • Half • None
Speed	The bandwidth of the interface.
LAG	The LAG ID of the port.
Tag	Indicates whether the ports have 802.1q VLAN tagging. The value can be Yes or No.
Priori	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 - 7.
MAC	The MAC address of the port.
Name	The name (if any) configured for the port.
Sys P	Lists the system priority configured for the device.
Port P	Lists the port's link aggregation priority.
Key	Lists the link aggregation key.
Act	Indicates the link aggregation mode, which can be one of the following: • No - The mode is passive on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes - The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L - Long. The LAG group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used as confirmation of the health of the aggregate link. • S - Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg - Link aggregation is enabled on the port. • No - Link aggregation is disabled on the port.
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No - The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a LAG link. • Syn - The port is in sync with the remote port. The port understands the status of the LACPDU message exchange

TABLE 21 Show LAG information (continued)

This field...	Displays...
	process, and therefore knows the LAG group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the LAG link: • Col - The port is ready to send traffic over the LAG link. • No - The port is not ready to send traffic over the LAG link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the LAG link. • Dis - The port is ready to receive traffic over the LAG link. • No - The port is not ready to receive traffic over the LAG link.
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. The port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a LAG group. An LACP port is blocked until it becomes part of a LAG. Also, an LACP is blocked if its state becomes "default". To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

Displaying LAG statistics

You can display LAG statistics for a Brocade device in either a **full** or **brief** mode. Full mode is the default and is displayed when the **show statistics lag** command is executed without the **brief** option. The examples below show both options of the **show statistics lag** command.

```
device# show statistics brief lag
LAG                  Packets            Collisions                  Errors
                      [Receive            Transmit]   [Recv  Txmit]      [InErr OutErr]
        LAG d1         1173              1018        0       0        0       0
         LAG e         1268              1277        0       0        0       0
device# show statistics lag
LAG d1 Counters:
         InOctets             127986           OutOctets             107753
```

```
          InPkts              1149           OutPkts               996
   InBroadcastPkts               0    OutBroadcastPkts                0
   InMulticastPkts             852    OutMulticastPkts              684
     InUnicastPkts             297      OutUnicastPkts              312
        InDiscards               0         OutDiscards                0
          InErrors               0           OutErrors                0
     InCollisions                0       OutCollisions                0
                                     OutLateCollisions                0
        Alignment                0                  FCS                0
        GiantPkts                0           ShortPkts                0
     InBitsPerSec                0       OutBitsPerSec                0
     InPktsPerSec                0       OutPktsPerSec                0
    InUtilization             0.0%      OutUtilization             0.0%
```

Available variations of this command include:

Syntax: show statistics [brief] lag [lag_name]

Syntax: show statistics brief lag

Syntax: show statistics brief lag lag_name

Syntax: show statistics lag

Syntax: show statistics lag lag_name

Displaying multicast LAG member port usage

Use the **show ip pim count lag** command to display the multicast LAG member port usage. The Forwarding entries correlate to the multicast (S, G) entries.

```
device# show ip pim count lag lag-member-port
PORT ID FORWARDING ENTRIES
e2/43 0
e2/44 0
e2/45 0
e2/46 0
```

Syntax: show ip pim count lag lag-member-port

For IPv6, use the **show ipv6 pim count lag** command.

Enter a LAG member port in the *lag-member-port* parameter.

Displaying LAG information for a specified LAG name or LAG ID

This **show interface lag** command displays LAG information of a LAG specified by the LAG name or LAG ID. Detailed information about each LAG interface, including counters, is displayed.

```
device# show interface lag lag1
Total number of LAGs: 1
Total number of deployed LAGs: 1
Total number of trunks created:1 (127 available)
LACP System Priority / ID: 1 / 0000.0001.c000
LACP Long timeout: 90, default: 90
LACP Short timeout: 3, default: 3
=== LAG "lag1" ID 123 (static Deployed) ===
LAG Configuration:
Ports: e 1/1 to 1/2
Port Count: 2
Primary Port: 1/1
Trunk Type: hash-based
Deployment: Trunk ID 123, Active Primary none, base fid: 0x0800
Port Link Port-State Dupl Speed Trunk Tag Priori MAC Name Type
1/1 DisabNone None None 123 No level0 0000.0001.c000
default-port
```

```
1/2 DisabNone None None 123 No level0 0000.0001.c000
default-port
LAG lag1 Counters:
InOctets 2237519128754 OutOctets 1050988054740
InPkts 1968838581 OutPkts 2030408443
InBroadcastPkts 0 OutBroadcastPkts 0
InMulticastPkts 0 OutMulticastPkts 0
InUnicastPkts 1968838581 OutUnicastPkts 2030448142
InDiscards 0 OutDiscards 0
InErrors 0 OutErrors 0
InCollisions 0 OutCollisions 0
OutLateCollisions 0
Sample Output Cont....
Alignment 0 FCS 0
GiantPkts 0 ShortPkts 0
InBitsPerSec 782177316 OutBitsPerSec 466226351
InPktsPerSec 90896 OutPktsPerSec 99992
InUtilization 7.96% OutUtilization 4.82%
```

Syntax: show interface lag lag-name

The *lag-name* or *lag ID* parameter can be used to display the detailed information of a specified the LAG. If no LAG name or LAG ID is specified, the detailed information of all the LAGs configured in the system will be displayed.

Displaying the running configuration for a LAG

The **show running-config lag** command displays the running configuration for a specified LAG or all LAGs as specified in the parameters.

```
device# show running-config lag detailed
!
lag "lag1" static id 1
ports ethernet 1/1
ports ethernet 1/2
ports ethernet 1/3
primary-port 1/1
deploy
!
lag "lag2" static id 2
ports ethernet 1/4
primary-port 1/4
```

Syntax: show running-config lag lag name

The *lag name* option displays the running configuration for the specified LAG. The *lag id* option may also be used to display the same information.

Use the *detailed* option to display the running-config on a specific *lag name* or *lag id*. If no LAG name or LAG id is specified, the information of the entire LAG configured in the system will be displayed.

Available variations of the command include:

Syntax: show running-config lag

Syntax: show running-config lag detailed

Syntax: show running-config lag detailed lag id

Syntax: show running-config lag detailed lag name

Syntax: show running-config lag lag id

Syntax: show running-config lag lag name

Displaying LACP information for a specified LAG name or LAG ID

Use the **show lacp** command to display LACP information for a specified LAG name or LAG ID. For each LAG port configured, the **show lacp** command displays the system identifier, system priority, port priority, and various state machine variables for both the actor and the partner of the system. The **show lacp** command also displays the LACP packets received on a port, LACP packets transmitted on a port, marker packets received on a port, and LACP error packets received on a port.The following example output displays LACP information for LAG ID 4.

> NOTE
> The **show lacp** command is supported on Brocade NetIron MLX Series, Brocade NetIron XMR Series, Brocade NetIron CER Series, and Brocade NetIron CES Series devices.

```
device#show lacp lag_id 4
[ACTR - ACTOR] [PRTR - PARTNER] [Act - Activity] [Tio - Timeout]
[Agg - Aggregation] [Syn - Synchronization] [Col - Collecting] [Dis - Distributing] [Def - Defaulted] [Exp
- Expired] [Ope - Operating]
=== LAG "e4-10g-1" ID 4 ===
Port Role Sys Port Oper [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope][Port]
          Pri Pri  Key                                                 Num

6/1  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   240
6/1  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   96
6/2  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   241
6/2  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope   97
6/3  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  242
6/3  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  146
6/4  ACTR  1   1  101    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  243
6/4  PRTR  1   1  240    Yes  L   Agg   Syn  Col  Dis   No   No  Ope  147
Actor     System MAC: 001b.ed04.3a00
Port  Partner                          LACP             LACP      LACP      MARKER
      System MAC           Rx Count    Tx Count  Err Count RX Count
6/1   0012.f2f7.3b00                    1495                1518       0        0
6/2   0012.f2f7.3b00          1496                                 1520       0        0
6/3   0012.f2f7.3b00          1497                                      1519             0        0
6/4   0012.f2f7.3b00          1499                                         1520       0        0
```

Syntax: show lacp [**lag_id** *number* | **lag_name** *name*]

The **lag_id** *number* parameter specifies the ID of the LAG you want to display.

The **lag_name** *name* parameter specifies the name of the LAG you want to display.

Use the **show lacp** command without any options to display LACP information for all dynamic or deployed LAGs configured on the system.

Table 22 displays the output information from the **show lacp** command.

TABLE 22 Output from the show lacp command

This Field...	Displays...
Port	Lists the port number of the LAG member. Displayed in slot/port format.
Role	Indicates if the LACP information displayed for a LAG port is for the actor or the partner.
System Priority (Sys Pri)	Lists the system priority configured for this port. The device with the lower system priority value takes the higher priority to be removed at the other end of the link. For example, a device with a system priority value of 1 has a higher priority to be removed than a device with a system priority value of 10. The system priority and System MAC address together form the system identifier of a LAG.
Port Priority (Port Pri)	Lists the priority value configured for this port. The port priority and the port number together form the port identifier of a LAG. The greater the port priority value, the greater the chances are for transmission. Data

TABLE 22 Output from the show lacp command (continued)

This Field...	Displays...
	traffic flow is distributed across multiple links on a LAG. The distribution of data traffic between links on a LAG is based on the port priority value. To configure the port priority value for a dynamic LAG or a keep-alive LAG, use the **lacp-port-priority** command. The priority value range is from 0 through 65535.
Operational Key (Oper key)	Lists the operational key value of a port. All ports on the LAG have the same key value. All ports with the same operational key value are aggregatable. The operational key value is assigned to the Ethernet link by the actor link. The key is dynamically generated based on the various port properties.
LACP_Activity (Act)	Indicates the control state of the link, which can be one of the following: • Yes - The link is active. The port can send and receive LACPDU messages. • No - The link is passive. The port does not initiate LACP messages, but will respond to LACP messages received.
LACP_Time (Tio)	Indicates the timeout value of the port. The timeout control value specifies the periodic transmission interval for LACP packets. The timeout control value can be set for a short timeout (3 seconds) or a long timeout (90 seconds). The LACP timeout control value is configurable using the **lacp-timeout** command. If the LACPDU is not received within the timeout mode configured on the port, the LACP will time out. If "S" is displayed, a short timeout value is used for the link. If "L" is displayed, a long timeout value is used for the link.
Aggregation (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.
Synchronization (Syn)	Indicates whether the link is allocated to the correct Link Aggregation Group. The link Aggregation Group is associated with a compatible aggregator to form the link aggregation. The identity of the group is consistent with the System ID and the operational key information that is transmitted. If "Syn" is displayed, the system considers this link to be IN_SYNC, and the link is allocated to the correct Link Aggregation Group. If "No" is displayed, the link is OUT_OF_SYNC, and the link is not in sync with aggregation.
Collecting (Col)	The collection of incoming frames that are enabled or disabled on the link. If "Col" is displayed, incoming frames is enabled on the link. If "No" is displayed, incoming frames is disabled on the link.
Distributing (Dis)	The distribution of outgoing frames that are enabled or disabled on the link. If "Dis"is displayed, the distribution of outgoing frames are enabled. If "No' is displayed, the distribution of outgoing frames are disabled.
Defaulted (Def)	Defaulted partner information is the set of LACP partner information (the system priority, key, port priority, and state of the partner) that is used when the information is not obtained from the partner through LACPDUs. This occurs when LACPDUs are not properly received on time. When "Def" is displayed, the actor's receive state machine is using the defaulted partner information that is configured administratively. If "No" is displayed, the actor's receive state machine is using the partner operational parameters that is received in a LACPDU.
Expired (Exp)	Indicates the state of the actor receive machine. If the LACP receive machine does not receive any LACPDUs within the timeout period configured on a port, the LACP receive machine goes into an EXPIRED state. The EXPIRED state indicates that the LAG has stopped operating.

TABLE 22 Output from the show lacp command (continued)

This Field...	Displays...
	When the actor receive machine starts receiving LACPDUs from the port at the other end of the link, it will begin operating again. When "Exp" is displayed, the actor receive machine is in the EXPIRED state. When "No" is displayed, the actor receive machine is operating and is not in the EXPIRED state.
Operating (Ope)	Indicates the operating status of the port. The port status can be one of the following: • Ope (operating) - 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 of the LAG is not configured with link aggregation. 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.
Port Number (Port Num)	The port number of the LAG.
Actor System MAC	The system MAC address of the actor. The system priority and system MAC address together form the system identifier.
Partner System MAC	The system MAC address of the partner. The system priority and system MAC address together form the system identifier.
LACP RX Count	The number of LACP packets received on a port.
LACP TX Count	The number of LACP packets sent on a port.
LACP Err Count	LACP is under the category of slow protocols. Slow protocol packets are received with an illegal subtype and a reserved subtype. LACP uses subtype 1, and the marker protocol uses subtype 2. Subtype values from 3 to 10 are reserved for future use. Subtype values 0 and 11 through 255 are considered illegal subtype values.
Marker RX Count	The number of marker packets received on a port. When a link is no longer aggregated with the port on the other end of the link, the marker protocol is used to verify that the conversation between the actor and the partner was successful on both ends. The actor and the partner exchange Marker PDUs and Marker Response PDUs to confirm the process.

Error messages displayed for LACP information when specifying a LAG name or LAG ID

If you do not configure a specified LAG name or LAG ID, an error message displays on the console, as shown in the following example.

```
device#show lacp lag_id 5
Error: LAG ID 5 is not configured
```

If you do not deploy a specified LAG name or LAG ID, an error message displays on the console, as shown in the following example.

```
device(config-lag-to-MLX2)#show lacp lag_id 1
Error: LAG 1 is not deployed
```

If you specify a LAG name or a LAG ID, and it is not a dynamic LAG, an error message displays on the console, as shown in the following example.

```
device(config-lag-abcd)#show lacp lag_id 4
Error: LAG 4 is not a dynamic LAG
```

Clearing LACP counter statistics for a specified LAG name or LAG ID

To clear LACP counter statistics for a specified LAG name or LAG ID, or for all LAGs in the system, enter the **clear lacp counters** command. The **clear lacp counters** command clears LACP packets that are received and transmitted on a LAG, in addition to clearing the LACP error count and the LACP marker packets that are received on all ports of the LAG.

> NOTE
> The **clear lacp counters** command is supported on Brocade NetIron MLX Series, Brocade NetIron XMR Series, Brocade NetIron CER Series and Brocade NetIron CES Series devices.

```
Brocade#clear lacp counters lag_id 1
```

Syntax: clear lacp counters [**lag_id** *number* | **lag_name** *name*]

The **lag_id** *number* parameter specifies the ID of the LAG for which you want to clear statistics.

The **lag_name** *name* parameter specifies the name of the LAG for which you want to clear statistics.

Use the **clear lacp counters** command without any options to clear all dynamic and deployed LAG statistics.

> NOTE
> Configuring a port as a member of an undeployed or deployed LAG resets LACP counter statistics to 0. Enabling or disabling a port does not clear LACP counters. After a switchover, LACP counter statistics display in the standby management module.

Brocade NetIron CES Series and Brocade NetIron CER Series Link Aggregation

This chapter describes how to configure Link Aggregation Groups (LAG) for Brocade NetIron CES Series and Brocade NetIron CER Series devices.

NOTE
The terms LAG and LAG groups are used interchangeably in this guide.

LAG formation rules

Multi-Service IronWare software supports the use a single interface to configure any of the following LAG types:

- **Static LAGs** - Manually-configured aggregate link containing multiple ports.
- **Dynamic LAG** - Uses the Link Aggregation Control Protocol (LACP), to maintain aggregate links over multiple ports. LACP PDUs are exchanged between ports on each device to determine if the connection is still active. The LAG then shuts down ports whose connection is no longer active. A syslog message is generated when the LAG is brought down because of he trunk-threshold being reached.
- **Keepalive LAG** - Establishes a single connection between a single port on 2 devices. LACP PDUs are exchanged between the ports to determine if the connection between the devices is still active. If it is determined that the connection is no longer active, the ports are blocked.

Follow these rules when configuring LAGs:

- You cannot configure a port concurrently as a member of a static, dynamic, or keepalive LAG
- Any number or combination of ports between 1 and 12 within the same device can be used to configure a LAG. The maximum number of LAG ports is checked when adding ports.
- All ports configured in a LAG must be of equal bandwidth. For example all 10 Gbps ports.
- All ports configured in a LAG must be configured with the same port attributes.
- LAG formation rules are checked when a static or dynamic LAG is deployed.
- A LAG must have the primary port selected before it can be deployed.
- All ports configured in a LAG must reside in the same VLAN.
- All dynamic LAG ports must have the same LACP BPDU forwarding configuration.

Layer 2 requirements

The LAG is rejected if the LAG ports:

- Do not have the same untagged VLAN component.
- Do not share the same VLAN membership and do not share the same uplink VLAN membership.
- Are configured as MRP primary and secondary interfaces.
- LAG deployment will fail if the LACP BPDU forwarding is disabled on the primary port and enabled on one or more of the secondary ports.

Layer 3 requirements

The LAG is rejected if any secondary LAG ports have any Layer 3 configuration, such as IPv4s, OSPF, RIP, RIPng, IS-IS, etc.

Layer 4 (ACL) requirements

- All LAG ports must have the same ACL configurations; otherwise, the LAG is rejected.
- A LAG cannot be deployed if a member port has ACL-based mirroring configured.
- A port with ACL-based mirroring configured cannot be added to a LAG.
- The device can support from 1-64 manually-configured LAGs.
- Ports can be in only one LAG group. All the ports in a LAG group must be connected to the same device at the other end. For example, if port 1/4 and 1/5 in Device 1 are in the same LAG group, both ports must be connected to ports in Device 2 or in Device 3. You cannot have one port connected to Device 2 and another port connected to Device 3.
- All LAG member properties must match the primary port of the LAG with respect to the following parameters:
  - Port tag type (untagged or tagged port)
  - Port speed and duplex
  - TOS-based configuration - All ports in the LAG must have the same TOS-based QoS configuration before LAG deployment, During deployment the configuration on the primary port is replicated to all ports and when deployment is ended, each port inherits the same TOS-based QoS configuration.

You must change port parameters on the primary port. The software automatically applies the changes to the other ports in the LAG.

- Make sure the device on the other end of the LAG group can support the same number of links in the LAG group.
- Dynamic LAGs are not supported for ports that are member of a VLAN within an ESI. Static LAGs are supported in such configurations.

Figure 6 displays an example of a valid, keepalive LAG link between two devices. A keepalive LAG does not aggregate ports but uses LACP PDUs to check the connection status between the two devices at either end of a LAG.

FIGURE 6 Example of a 1-port keepalive LAG

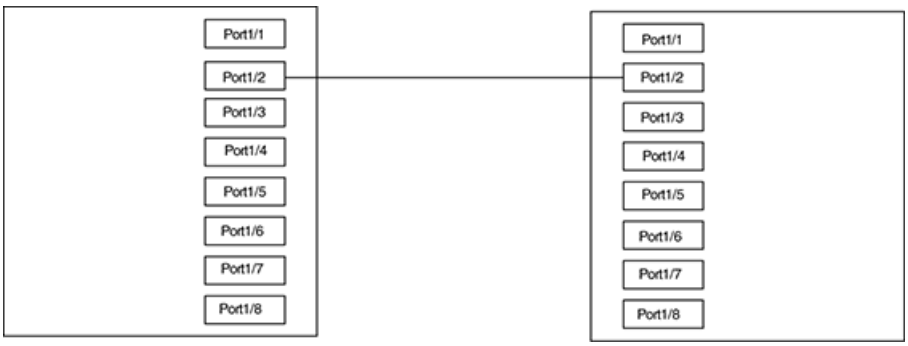
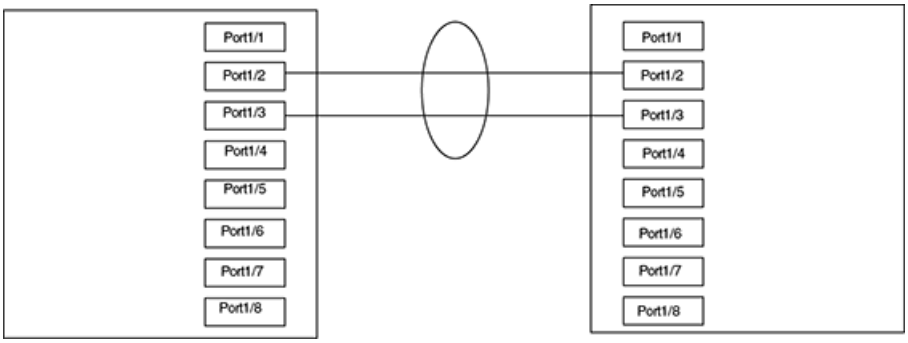


Figure 7 shows an example of a valid 2-port LAG between devices where the ports on each end are on the same interface module. Ports in a valid 2-port LAG on one device are connected to two ports in a valid 2-port LAG on another device.

FIGURE 7 Example of 2-port LAG



LAG load sharing

Brocade devices can be configured for load sharing over a LAG using hash-based load sharing.

Hash based load sharing

Brocade NetIron CES Series and Brocade NetIron CER Series devices share the traffic load evenly across the ports in LAG group, while ensuring that packets in the flow are not reordered. Individual flows are assigned a LAG index to identify them.

Traffic from each flow is then distributed across the ports in the LAG group using a hash index as follows:

The hash value is calculated based on the packet:

- hash[5:0] = MAC_SA[5:0]^MAC_DA[5:0];

If the packet is IP (v4 or v6), the hash is further calculated as the following:

- hash[5:0] = hash[5:0]^SIP[5:0]^SIP[21:16]^DIP[5:0]^DIP[21:16];

If the packet is TCP or UDP, the hash is further calculated as the following:

- hash[5:0] = hash[5:0]^L4_SrcPort[5:0]^L4_SrcPort[13:8]^L4_TrgPort[5:0]^L4_TrgPort[13:8];

If the packet is not IP:

- If the packet is MPLS, the hash is further calculated as the following:
  - hash[5:0] = hash[5:0]^MPLS_L0[5:0]^MPLS_L1[5:0]^MPLS_L2[5:0];
- The 4-bit hash value is: hash[3:0] = hash[5:0]% num_of__members;

NOTE
The hashing in CES/CER is based on the values present in the header of the incoming traffic (L2, L3, and L4). If the incoming traffic is equal to the total bandwidth of the outgoing LAG, there is no guarantee that the packet will be equally shared among the LAG links.

Not supported:

- Speculate UDP or TCP headers
- Mask Layer-4 source
- Hash diversification

Deploying a LAG

After configuring a LAG, you must explicitly enable it using the **deploy** command before it begins aggregating traffic. Once the **deploy** command is executed, the LAG enters aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG.

To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keepalive LAGs. Once a non keepalive LAG is deployed, a LAG is formed. If there is only one port in the LAG, a single-port LAG is formed. For a dynamic LAG, LACP is started for each LAG port. For a keepalive LAG, no LAG is formed and LACP is started on the LAG port.

Refer to the *Brocade NetIron Administration Guide* for additional information on LAG naming conventions.

You can deploy a LAG as shown for the "blue" LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
```

Syntax: [no] deploy [**forced** | **passive**]

When the **deploy** command is executed:

- For a static and dynamic LAGs, the LAG veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a LAG is formed with all the ports in the LAG.
- For dynamic LAGs, LACP is activated on all LAG ports. When activating LACP, use **active** mode if **passive** is not specified; otherwise, use **passive** mode.
- For keepalive LAGs, no LAG is formed, and LACP is started on the LAG port.
- Once the **deploy** command is issued, all LAG ports will behave like a single port.
- If the **no deploy** command is executed, the LAG is removed. For dynamic LAGs, LACP is deactivated on all LAG ports.
- If the **no deploy** command is issued and more than 1 LAG port is not disabled, the command is aborted and the following error message is displayed: "Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted." Using the **forced** keyword with the **no deploy** command in the previous situation, the LAG deployment is cancelled.

Commands available under LAG once it is deployed

Once a LAG has been deployed, the following configurations can be performed:

- Configuring ACL-based Mirroring
- Disabling Ports within a LAG
- Enabling Ports within a LAG
- Monitoring and Individual LAG Port
- Assigning a name to a port within a LAG
- Enabling sFlow Forwarding on a port within a LAG
- Setting the sFlow Sampling Rate for a port within a LAG

Configuring ACL-based mirroring

To configure ACL-based mirroring for all ports in a LAG, configure it on the primary port of the LAG at the interface configuration level (see Configuring IP Chapter). ACL-based mirroring can be configured for an individual member port within a LAG by using the **acl-mirror-port** command.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# acl-mirror-port ethe-port-monitored 3/1 ethernet 3/2
```

In this example, traffic on Ethernet port 3/1 (a secondary member port of LAG "blue") will be mirrored to Ethernet port 3/2.

Syntax: [no] acl-mirror-port { **ethe-port-monitored** *slot/port* | **named-port-monitored** *name* } **ethernet** *slot/port*

Use the **ethe-port-monitored** option with the appropriate *slot/port* variable to specify an Ethernet port for which you want to provide ACL mirroring.

Use the **named-port-monitored** option with the appropriate *slot/port* variable to specify a named port for which you want to provide ACL mirroring.

The **ethernet** keyword precedes the *slot/port* variable, identifying the port which will receive the mirrored packets.

A port with ACL-based mirroring already configured cannot be added to a LAG, and a LAG cannot be deployed if any member ports are configured for ACL-based mirroring. To use ACL-based mirroring on a LAG member port, deploy the LAG, then configure mirroring on the member port. If a port is removed from a LAG, ACL-based mirroring is removed from that port, and if a LAG is deleted mirroring is removed from all member ports.

Disabling ports within a LAG

You can disable an individual port within a LAG using the **disable** command within the LAG configuration.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# disable ethernet 3/1
```

Syntax: [no] disable ethernet [*slot/port*] | **named** *name*

Use the **ethernet** option with the appropriate *slot/port* variable to specify a Ethernet port within the LAG that you want to disable.

Use the **named** option with the appropriate *slot/port* variable to specify a named port within the LAG that you want to disable.

When a port is deleted from a deployed static LAG, the LACP BDPU forwarding state of the LAG will be retained for the deleted port.

Enabling ports within a LAG

You can enable an individual port within a LAG using the **enable** command.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# enable ethernet 3/1
```

Syntax: [no] enable ethernet [*slot/port*] | **named** *name*

Use the **ethernet** option with the appropriate *slot/port* variable to specify an Ethernet port to be enabled in the LAG.

Use the **named** option with the appropriate *slot/port* variable to specify a named port in the LAG that you want to enable.

When adding a port to a currently deployed dynamic LAG the LACP BPDU Forwarding configuration must be the same as the LAG. Follow the procedure Enabling and Disabling LACP BPDU Forwarding on a Port on page 157.

Monitoring an individual LAG port

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

You can use only one mirror port for each monitored LAG port. To monitor traffic on an individual port in a LAG group, enter commands such as the following:

This command enables monitoring of an individual port within a LAG.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# monitor ethe-port-monitored 3/1 ethernet 10/3 both
```

Syntax: [no] monitor ethe-port-monitored *slot/port* | **named-port-monitored** *name* | **ethernet** *slot/port* [**input** | **output** | **both**]

Use the **ethe-port-monitored** option with the appropriate *slot/port* variable to specify an Ethernet port in the LAG that you want to monitor.

Use the **named-port-monitored** option with the appropriate *slot/port* variable to specify a named port in the LAG that you want monitor.

The **ethernet** *slot/port* parameter specifies the port to which the traffic analyzer is attached.

The **input, output,** and **both** parameters specify the traffic direction to be monitored.

Naming a port in a LAG

You can name an individual port in a LAG using the **port-name** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# port-name orange ethernet 3/1
```

Syntax: [no] port-name text ethernet [**slot/port**]

The *text* variable specifies the port name. The name can be up to 50 characters long.

Use the **ethernet** option with the appropriate *[slot/port]* variable to apply the specified name to an Ethernet port within the LAG.

Refer to the *Brocade NetIron Administration Guide* for additional information on LAG naming conventions.

NOTE
The port name and LAG name cannot use the same name.

Enabling sFlow forwarding on a port in a LAG

You can enable sFlow forwarding on an individual port within a LAG using the **sflow-forwarding** command within the LAG configuration as shown in the following.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-forwarding ethernet 3/1
```

Syntax: [no] sflow-forwarding ethernet *slot/port* | **port-name** *name*

Use the **ethernet** option with the appropriate *slot/port* variable to specify an Ethernet port in the LAG where you want to enable sFlow forwarding.

Use the **port-name** option with the appropriate *name* variable to specify a named port within the LAG where you want to enable sFlow forwarding.

Setting the sFlow sampling rate for a port in a LAG

NOTE
The NetIron CES and NetIron CER supports sflow sampling rate configuration per port basis. The Brocade MLXe, NetIron MLX, and NetIron XMR supports sflow sampling rate configuration per packet processer basis.

You can set the sFlow sampling rate for an individual port within a LAG using the **sflow-subsampling** command as shown.

```
device(config)# lag blue static
device(config-lag-blue)# deploy
device(config-lag-blue)# sflow-subsampling ethernet 3/1 512
```

Syntax: [no] sflow-subsampling *num*

The *num* variable specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. This can be a value between 512 - 1048576.

Static LAG Considerations

Enabling and Disabling LACP BPDU Forwarding on a Port

NOTE
The forward-lacp command must be issued on the physical port configuration, not in LAG configuration.

For scenarios in which dynamic LAG ports require LACP BPDU packet forwarding, you can issue the **forward-lacp** command in the interface mode. Once LACP Forwarding has been enabled on a dynamic LAG, all the LACP BPDUs will follow regular packet forwarding actions.

When LACP forwarding is enabled, the link OAM packets received on the LACP forwarding enabled interface will be processed and flooded on the VLAN. If the LACP forwarding is not enabled, the link OAM packets will be processed and then dropped.

To enable LACP BPDU forwarding, enter the following command:

```
device(config-if-e1000-3/5)# forward-lacp
```

To disable LACP BPDU forwarding, enter the lacp-forwarding command as follows.

```
device(config-if-e1000-3/5)# [no]
forward-lacp
```

Enabling and Disabling LACP BPDU Forwarding on a Trunk

NOTE
The forward-lacp command must be issued on the physical port configuration, not in LAG configuration.

When the LACP forwarding is enabled on the primary port of the dynamic LAG, the LACP BPDU forwarding is enabled on all ports of the LAG when the LAG is deployed. When the static LAG is undeployed the BPDU forwarding state is retained.

- If LACP BPDU forwarding is enabled on the primary and secondary ports, the LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is enabled on the primary port and disabled on the secondary ports, the LAG deployment will be successful and LACP BPDU forwarding will be enabled on the LAG ports.
- If LACP BPDU forwarding is disabled on the primary and secondary ports, the LAG deployment will be successful and LACP BPDU forwarding will be disabled on the LAG ports.

NOTE
LACP BPDU forwarding is not supported for any port of dynamic or keep alive LAGs.

Displaying LAG information

You can display LAG information for by entering the **show lag** command.

```
device# show lag
Total number of LAGs:          4
Total number of deployed LAGs: 3
Total number of s created:3 (125 available)
LACP System Priority / ID:     0001 / 0004.80a0.4000
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "d1" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 13/2 to 13/3 ethe 32/2
   Primary Port:  32/2
    Type:    hash-based
   LACP Key:      104
Deployment:   ID 3, Active Primary 3/2
Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
3/2   Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
13/3  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
32/2  Up   Forward   Full 10G   3     Yes level0 0004.80a0.44d9
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
13/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
13/3       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
32/2       1        1     104   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
=== LAG "e" (dynamic Deployed) ===
LAG Configuration:
   Ports:         ethe 2/1 ethe 2/3 ethe 2/5
   Primary Port:  2/3
    Type:    hash-based
   LACP Key:      105
Deployment:   ID 1
Port  Link L2 State  Dupl Speed  Tag Priori MAC            Name
2/1   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/3   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
2/5   Up   Forward   Full 1G    1     Yes level0 0004.80a0.402a
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
2/1        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/3        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
2/5        1        1     105   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
```

Syntax: show lag *lag-name* [**brief**] [**deployed**] [**dynamic**] [**ID**] [**Ethernet**] [**keepalive**] [**static**]

Using this command without options displays information for all LAGs configured on the device. In addition, the following arguments may be used to provide additional LAG information.

The *lag-name* variable allows you to limit the display to information for a specific LAG.

The **ID** option will display the **show lag** command output for the LAG specified by the ID.

The **brief** option displays summary information for any or all configured LAGs.

The **deployed** option limits the display to LAGs that are currently deployed.

The **dynamic** option limits the display to dynamic LAGs.

The **Ethernet** option displays the output for the specified Ethernet port.

The **keep-alive** option limits the display to keep alive LAGs.

The **deployed** option limits the display to static LAGs.

Table 23 describes the information displayed by the **show lag** command.

TABLE 23 Show LAG information

This field...	Displays...
Total number of LAGS	The total number of LAGs that have been configured on the device.

TABLE 23 Show LAG information (continued)

This field...	Displays...
Total number of Deployed LAGS	The total number of LAGs on the device that are currently deployed.
Total number of LAGs Created	The total number of LAGs that have been created on the LAG. The total number of LAGs available are shown also. Since keepalive LAGs do not use an ID, they are not listed and do not subtract for the number of LAGs available.
LACP System Priority /ID	The system priority configured for the device.The ID is the system priority which is the base MAC address of the device.
LACP Long timeout	The number of seconds used for the LACP Long timeout mode. This is only applicable for dynamic or keepalive LAGs.
LACP Short timeout	The number of seconds used for the LACP Short timeout mode. This is only applicable for dynamic or keepalive LAGs.
The following information is displayed per-LAG in the show lag brief command.	
LAG	The name of the LAG.
Type	The configured type of the LAG: static, dynamic, or keepalive
Deploy	Status of LAG deployment: Y - yes, LAG is deployed. N - no, LAG is not deployed.
LAG	The LAG ID number.
Primary	The primary port of the LAG.
Port List	The list of ports that are configured in the LAG.
The following information is displayed per-LAG the show lag command for each LAG configured.	
LAG Configuration	
Ports:	List of ports configured in the LAG.
Primary Port:	The primary port for the LAG.
LAG Type:	The load sharing method configured for the LAG: hash-based.
LACP Key	The link aggregation key for the LAG.
Deployment	
LAG ID	The LAG ID number.
Active Primary	The LAG port where most protocol packets are transmitted. This is not the same as the configured Primary Port of the LAG.
Port	The chassis slot and port number of the interface.
Link	The status of the link, which can be up or down.
L2 State	The Layer 2 state for the port.
Dupl	The duplex state of the port, which can be one of the following: • Full • Half • None
Speed	The bandwidth of the interface.
LAG	The LAG ID of the port.
Tag	Indicates whether the ports have 802.1q VLAN tagging. The value can be Yes or No.
Priori	Indicates the Quality of Service (QoS) priority of the ports. The priority can be a value from 0 - 7.
MAC	The MAC address of the port.

TABLE 23 Show LAG information (continued)

This field...	Displays...
Name	The name (if any) configured for the port.
Sys P	The system priority configured for the device.
Port P	Link aggregation priority of the port
Key	Lists the link aggregation key.
Act	Indicates the link aggregation mode, which can be one of the following: • No - The mode is passive on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes - The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L - Long. The LAG group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used to confirm the health of the aggregate link. • S - Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg - Link aggregation is enabled on the port. • No - Link aggregation is disabled on the port.
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No - The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a LAG link. • Syn - The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and knows the LAG group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the LAG link: • Col - The port is ready to send traffic over the link. • No - The port is not ready to send traffic over the link.
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the LAG link: • Dis - The port is ready to receive traffic over the LAG link. • No - The port is not ready to receive traffic over the LAG link.
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.

TABLE 23 Show LAG information (continued)

This field...	Displays...
	• No - The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp - The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No - The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a LAG group. An LACP port is blocked until it becomes part of a LAG group. 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.

To display long port names, **set-lag-port-mode-wid** command. This command is useful if the ports names are long. In wide mode, the complete port name will be displayed and port type will not be displayed. In standard (non-wide) mode, only a portion of the port name is displayed if the port name is long and the port type is displayed. The following example shows the wide-mode display of a **show lag** command.

: Wide-Mode Display

```
device(config)#set-lag-port-mode-wid
device((config)#sh lag
Total number of LAGs:          2
Total number of deployed LAGs: 2
Total number of trunks created:2 (126 available)
LACP System Priority / ID:     1 / 00da.1111.2200
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "1234567890$%'-_@~`!(){}^#&abcdefghijklmnopqrstuvwXYZ" ID 4 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/3 to 31/10 e 31/15 to 31/20
   Port Count:    14
   Primary Port:  31/3
   Trunk Type:    hash-based
   LACP Key:      101
Port Individual Configuration:
   Port Name
   31/3 test2
Deployment:  Trunk ID 4, Active Primary none, base fid: 0x0810
Port  Link Port-State  Speed Tag MAC            Name
31/3  DisabNone        None  No  00da.1111.27a2 test2
31/4  DisabNone        None  No  00da.1111.27a2
31/5  DisabNone        None  No  00da.1111.27a2
31/6  DisabNone        None  No  00da.1111.27a2
31/7  DisabNone        None  No  00da.1111.27a2
31/8  DisabNone        None  No  00da.1111.27a2
31/9  DisabNone        None  No  00da.1111.27a2
31/10 DisabNone        None  No  00da.1111.27a2
31/15 DisabNone        None  No  00da.1111.27a2
31/16 DisabNone        None  No  00da.1111.27a2
31/17 DisabNone        None  No  00da.1111.27a2
```

```
31/18 DisabNone        None  No  00da.1111.27a2
31/19 DisabNone        None  No  00da.1111.27a2
31/20 DisabNone        None  No  00da.1111.27a2
=== LAG "NN" ID 6 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 31/11 to 31/14 e 31/21 to 31/24
   Port Count:    8
   Primary Port:  31/11
   Trunk Type:    hash-based
   LACP Key:      100
Port Individual Configuration:
   Port Name
   31/11test
Deployment:  Trunk ID 6, Active Primary 31/12, base fid: 0x0800
Port  Link Port-State  Speed Tag MAC            Name
31/11 Up   Forward     1G    No  00da.1111.27aa test
31/12 Up   Forward     1G    No  00da.1111.27aa
31/13 Up   Forward     1G    No  00da.1111.27aa
31/14 Up   Forward     1G    No  00da.1111.27aa
31/21 Up   Forward     1G    No  00da.1111.27aa
31/22 Up   Forward     1G    No  00da.1111.27aa
31/23 Up   Forward     1G    No  00da.1111.27aa
31/24 Up   Forward     1G    No  00da.1111.27aa
```

Syntax: [no] set-lag-port-mode-wid

Displaying LAG statistics

You can display LAG statistics in either **full** or **brief** mode. Full mode is the default and is displayed when the **show statistics lag** command is executed without the **brief** option. These examples show both options of the **show statistics lag** command.

```
device# show statistics brief lag
LAG                   Packets            Collisions                   Errors
                       [Receive            Transmit]   [Recv  Txmit]      [InErr OutErr]
        LAG d1          1173              1018        0       0        0       0
         LAG e          1268              1277        0       0        0       0
device# show statistics lag
LAG d1 Counters:
         InOctets             127986           OutOctets             107753
           InPkts               1149             OutPkts                996
  InBroadcastPkts                 0    OutBroadcastPkts                  0
  InMulticastPkts               852    OutMulticastPkts                684
    InUnicastPkts               297      OutUnicastPkts                312
       InDiscards                 0         OutDiscards                  0
         InErrors                 0           OutErrors                  0
     InCollisions                 0       OutCollisions                  0
                                      OutLateCollisions                  0
        Alignment                 0                 FCS                  0
        GiantPkts                 0           ShortPkts                  0
     InBitsPerSec                 0       OutBitsPerSec                  0
     InPktsPerSec                 0       OutPktsPerSec                  0
    InUtilization              0.0%      OutUtilization               0.0%
```

Syntax: show statistics [**brief**] **lag** [*lag-name*]

The following syntax options can be used for a brief display:

Syntax: show statistics brief lag

Syntax: show statistics brief lag *lagname*

The following syntax options can be used for a full display:

Syntax: show statistics lag

Syntax: show statistics lag *lagname*

Displaying LAG information for a specified LAG name or LAG ID

The **show interfaces lag** command displays LAG information for a LAG specified by LAG name or LAG ID. If no LAG name or LAG ID is specified, it shows detailed information of all the LAGs configured in the system. For each port of a LAG, detailed information about the LAG interface, including counters is displayed.

```
device#show interfaces lag
Total number of LAGs:          2
Total number of deployed LAGs: 2
Total number of trunks created:2 (62 available)
LACP System Priority / ID:     1 / 001b.edb3.f181
LACP Long timeout:             90, default: 90
LACP Short timeout:            3, default: 3
=== LAG "151-188" ID 2 (static Deployed) ===
LAG Configuration:
   Ports:         e 1/3 to 1/4 e 1/15 to 1/16 e 1/27 e 1/39 to 1/40
   Port Count:    7
   Primary Port:  1/3
   Trunk Type:    hash-based
   LACP BPDU Forwarding: Disabled
Deployment:  Trunk ID 2, Active Primary 1/3, base fid: 0x0000
Port  Link Port-State  Dupl Speed Trunk Tag Priori MAC            Name           Type
1/3   Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/4   Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/15  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/16  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/27  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/39  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
1/40  Up   Forward     Full 1G    2     Yes level0 001b.edb3.f183               default-port
LAG 151-188 Counters:
         InOctets          71899883          OutOctets            6816239
           InPkts            865449            OutPkts              23691
  InBroadcastPkts             25684    OutBroadcastPkts                  0
  InMulticastPkts            839765    OutMulticastPkts              23691
    InUnicastPkts                 0      OutUnicastPkts                  0
       InDiscards                 0         OutDiscards                  0
         InErrors                 0           OutErrors                  0
     InCollisions                 0       OutCollisions                  0
                                      OutLateCollisions                  0
        Alignment                 0                 FCS                  0
        GiantPkts                 0           ShortPkts                  0
     InBitsPerSec              7846       OutBitsPerSec                  0
     InPktsPerSec                10       OutPktsPerSec                  0
    InUtilization              0.0%      OutUtilization               0.0%
=== LAG "151-189" ID 1 (dynamic Deployed) ===
LAG Configuration:
   Ports:         e 1/1 to 1/2 e 1/13 to 1/14 e 1/25 to 1/26 e 1/37 to 1/38
   Port Count:    8
   Primary Port:  1/1
   Trunk Type:    hash-based
   LACP Key:      100
Deployment:  Trunk ID 1, Active Primary 1/1, base fid: 0x0000
Port  Link Port-State  Dupl Speed Trunk Tag Priori MAC            Name           Type
1/1   Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/2   Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/13  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/14  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/25  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/26  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/37  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
1/38  Up   Forward     Full 1G    1     Yes level0 001b.edb3.f181               default-port
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1       1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/2       1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/13      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/14      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/25      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/26      1        1     100   Yes   L   Agg  Syn  Col  Dis  No   No   Ope
1/
LAG 151-189 Counters:
```

```
        InOctets          13357835          OutOctets          73729556
          InPkts             54050            OutPkts            885774
  InBroadcastPkts                 0    OutBroadcastPkts             25682
  InMulticastPkts             54050    OutMulticastPkts            860092
    InUnicastPkts                 0      OutUnicastPkts                 0
       InDiscards                 0         OutDiscards                 0
        InErrors                  0           OutErrors                 0
     InCollisions                 0       OutCollisions                 0
                                      OutLateCollisions                 0
        Alignment                 0                 FCS                 0
        GiantPkts                 0           ShortPkts                 0
     InBitsPerSec                 0       OutB
```

Syntax: show interfaces lag *lag-name*

The *lag-name* or *lag ID* parameter can be used to display the detailed information of a specified the LAG.

Displaying the running configuration for a LAG

The **show running-config lag** command displays the running configuration for a specified LAG or all LAGs as specified in the parameters.

```
device# show running-config lag detailed
!
lag "lag1" static id 1
ports ethernet 1/1
ports ethernet 1/2
ports ethernet 1/3
primary-port 1/1
deploy
!
lag "lag2" static id 2
ports ethernet 1/4
primary-port 1/4
```

Syntax: show running-config lag *lagname*

The *lag name* option displays the running configuration for the specified LAG. The *lag id* option may also be used to display the same information.

Use the **detailed** option to display the running-config on a specific *lag name* or *lag id*. If no LAG name or LAG id is specified, the information of the entire LAG configured in the system will be displayed.

The following command options may be used:

Syntax: show running-config lag

Syntax: show running-config lag detailed

Syntax: show running-config lag detailed *lag_id*

Syntax: show running-config lag detailed *lagname*

Syntax: show running-config lag *lag_id*

Syntax: show running-config lag *lagname*

VLANs

A Virtual Local Area Network (VLAN) lets you segment traffic in a network by placing ports and interfaces into separate broadcast domains. Each broadcast domain is uniquely identified by a VLAN ID. These broadcast domains can span multiple devices.

NOTE

The Brocade NetIron CES Series devices support the Ethernet Service Instance (ESI) framework. A user can configure ESIs in the process of configuring Provider Bridges and Provider Backbone Bridging. By default, the device has a "default ESI" configured in which VLANs 1 - 4090 exist. This chapter refers to configuration and use VLANs under the default ESI framework. For configuration of user-defined ESIs, please refer to the ESI framework, which is described in the *Ethernet Service Instance (ESI) for Brocade NetIron CES and Brocade NetIron CER devices* chapter.

The Brocade device supports two types of VLANs: *port-based VLANs* and *protocol-based VLANs* . A port-based VLAN consists of interfaces that constitute a Layer 2 broadcast domain. (Protocol-based VLANs are described in Protocol-based VLANs on page 167.) By default, all interfaces on a Brocade device are members of the *default* VLAN, which is VLAN 1. Thus, by default, all interfaces on all devices on a network constitute a single Layer 2 broadcast domain. Once you create a port-based VLAN and assign an interface to that

VLAN, that interface is automatically removed from the default VLAN if the interface is assigned to the VLAN as an untagged interface. If the interface is assigned as a tagged interface, then the interface is a member of both the default VLAN, and the VLAN to which it is assigned.

Tagged, untagged, and dual mode ports

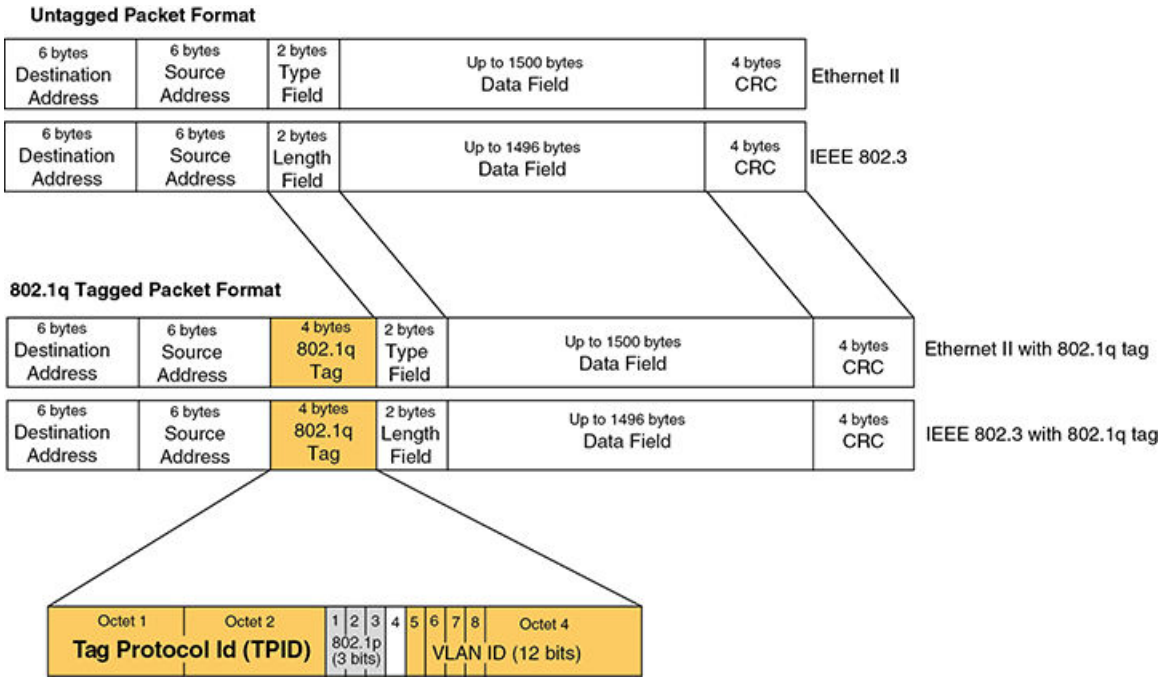
Interfaces assigned to port-based VLANs can be defined as untagged, tagged, and dual-mode ports. An untagged port is a member of only one VLAN, while a tagged port can be a member of more than one VLAN. Thus a tagged port can be a member of more than one broadcast domain. Dual-mode ports are configured by adding one or more tagged VLANs and one untagged VLAN to a port.

Tagged ports allow the Brocade device to add a four-byte 802.1q tag to the packet. 802.1q tagging is an IEEE standard that allows a networking device to add information to Layer 2 packets. This information identifies the VLAN membership of the packet, as well as the VLAN ID of the VLAN from which the packet is sent. Furthermore, the default tag value of the 802.1q tag is 8100 (hexadecimal). This value comes from the IEEE 802.1q specification. You can change this tag value on a per-port or on a global basis on a Brocade device if needed to be compatible with other vendors' equipment.

> NOTE
> On Brocade NetIron CES Series devices, you can change the tag value on the global basis for each VLAN component (B-VLAN, C-VLAN, or S-VLAN).

Figure 8 shows the format of packets with and without the 802.1q tag.

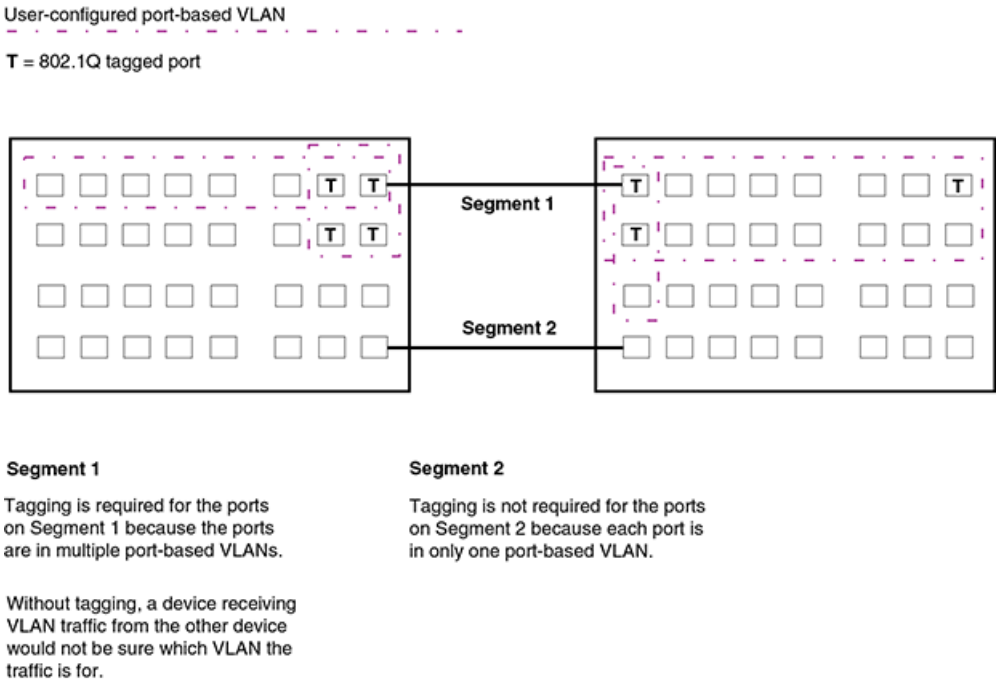
FIGURE 8 Packet containing Brocade's 802.1QVLAN tag



If you configure a VLAN that spans multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If a port connecting one device to the other is a member of only a single port-based

VLAN, tagging is not required. Figure 9 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 VLANs configured across multiple devices



Protocol-based VLANs

Interfaces that belong to a port-based VLAN can further be divided into Layer 3 broadcast domains by using protocol-based VLANs. Protocol-based VLANs accept broadcasts of a specified protocol type. For example, an IP subnet VLAN accepts broadcasts for the specified IP subnets only. This feature enables you to limit the amount of broadcast traffic to end-stations, servers, and routers.

In a Brocade device, you can configure the following protocol-based VLANs within a port-based VLAN:

- **AppleTalk** - The device sends AppleTalk broadcasts to all ports within the AppleTalk protocol VLAN.
- **IP** - The device sends IP broadcasts to all ports within the IP protocol VLAN.
- **IPX** - The device sends IPX broadcasts to all ports within the IPX protocol VLAN.
- **IPv6** - The device sends IPv6 broadcasts to all ports within the IPv6 protocol VLAN.

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

Protocol-based VLANs can be configured to have *static* or *excluded* port memberships. Static ports are permanent members of a protocol-based VLAN. They remain active members of the protocol-based VLAN regardless of whether they receive traffic for the VLAN's protocol.

NOTE
The dynamic port membership is not supported on Brocade devices.

If you want to exclude certain ports in a port-based VLAN from protocol-based VLANs, the protocol-based VLAN can be explicitly configured to exclude those ports.

VLAN configuration rules

To create any type of VLAN on a Brocade router, Layer 2 forwarding must be enabled. When Layer 2 forwarding is enabled, the Brocade device becomes a switch on all ports for all non-routable protocols.

In addition to this rule, the sections below summarize the rules for configuring VLANs.

NOTE
To enable Layer 2 forwarding, use the **no route-only command.** On Brocade NetIron CES Series devices, Layer 2 forwarding is enabled by default.

VLAN ID range

The upper range of VLAN IDs available for user VLANs (including the default VLAN) has been reduced to 4090 (formerly 4094). The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

Tagged VLANs

When configuring VLANs across multiple devices, you need to use tagging only if a port connecting one of the devices to the other is a member of more than one port-based VLAN. If you are configuring tagged VLANs across multiple devices, make sure all the devices support the same tag format.

VLAN hierarchy

A hierarchy of VLANs exists between the Layer 2 and Layer 3 protocol-based VLANs:

- Port-based VLANs are at the lowest level of the hierarchy.
- Layer 3 protocol-based VLANs are at the highest level of the hierarchy.

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

When a port in a VLAN receives a packet, the device forwards the packet based on the following VLAN hierarchy:

- If it is a Layer 3 packet and the port is a member of a Layer 3 protocol-based VLAN for the packet's protocol, the device forwards the packet on all the Layer 3 protocol-based VLAN ports that have been configured or drops the packet if the port is explicitly excluded from the protocol VLAN.
- If the packet cannot be forwarded based on its VLAN membership types but the packet can be forwarded at Layer 2, the device forwards the packet on all the ports within the receiving port's port-based VLAN.

Multiple VLAN membership rules

The multiple VLAN membership rules are listed below:

- A port can belong to multiple, overlapping Layer 2 port-based VLANs only if the port is a tagged port. Packets sent out of a tagged port use an 802.1q-tagged frame.
- A port can belong to multiple, unique, overlapping Layer 3 protocol-based VLANs.
- When both port and protocol-based VLANs are configured on a given device, all protocol-based VLANs must be strictly contained within a port-based VLAN. A protocol-based VLAN cannot include ports from multiple port-based VLANs. This rule is required to ensure that port-based VLANs remain loop-free Layer 2 broadcast domains.
- One of each type of protocol-based VLAN can be configured within each port-based VLAN on the Brocade device.
- Removing a configured port-based VLAN from a Brocade device automatically removes any protocol-based VLAN, or any virtual routing interfaces defined within the port-based VLAN.

Dual-mode default VLAN

As previously described, ports can be defined as dual-mode, which means that they can exist in both tagged and untagged VLANs. As such, they can coexist untagged in the default or a non-default VLAN and be added as a tagged port into non-default VLAN. One way that ports become dual-mode is by adding a port to a non-default, tagged VLAN. The normal behavior is for the port to remain in the default VLAN as an untagged port.

Changing the dual-mode default VLAN behavior

The **no dual-mode-default-vlan** command has been added to change this behavior. This is useful in situations where there is a danger of loops being created if Spanning Tree is not or can not be configured on the default VLAN such as when ports are facing a service provider network and STP BPDUs are not welcome on those ports.

Once the **no dual-mode-default-vlan** command is applied at the global level, a port will not be entered into the dual-mode state by default. If the **no dual-mode-default-vlan** command is configured, when a port is added as tagged to a non-default user-defined VLAN, it is automatically removed from the default VLAN and added to the non-default VLAN as a pure tagged port. Once in this state, a port can only be placed in dual-mode by explicitly configuring it as an untagged port into a non-default VLAN.

When the **no untagged ethernet** command is applied under the default VLAN against a port in dual mode, the port will go into pure tagged mode in contrast to the default operating conditions where the port is automatically placed in the dual mode state with regard to the default VLAN. To change the default condition of a Brocade device regarding the dual-mode, default VLAN behavior, enter the **no dual-mode-default-vlan** command as shown in the following.

```
device(config)# no dual-mode-default-vlan
```

Syntax: [no] dual-mode-default-vlan

The default state is for ports added as tagged to a non-default VLAN to remain as untagged ports in the default VLAN and become dual-mode ports.

Using this command with the **no** option, changes the default state and automatically removes a port from the default VLAN when it is added as a tagged port to a non-default VLAN. Using the command without the **no** option, will return the systems behavior to its normal operating condition.

NOTE

When a Brocade device is operating in the default state regarding the **no dual-mode-default-vlan** command (which is not configured), syslog messages are generated whenever a port is moved out of the default VLAN. This is normal and expected behavior. Because when the **no dual-mode-default-vlan** command is configured, it is normal operating behavior for a port to be moved out of the default VLAN whenever it enters the dual-mode state syslog messages may be generated when the ports are moved, which is not expected. These messages may be generated in the following situations:

- When a port is added "tagged" into the default VLAN, it is automatically deleted from the default VLAN and a syslog message is generated.
- When a port is in dual-mode, and a user issues the **no untagged** command within the port VLAN configuration, the port is added back to the default VLAN. However, because the **no dual-mode-default-vlan** command is configured, the port is transitioned out of the default VLAN which generates an additional syslog message.

Restrictions for use of this command

The **no dual-mode-default-vlan** command can only be applied if the device does not currently have any ports configured in dual-mode. If any port is currently in the dual-mode state when the **no dual-mode-default-vlan** command is executed, the command is rejected without it being applied to any ports on the device. Consequently, using the **no dual-mode-default-vlan** command does not cause any action but enables a new behavior for ports that are added to a VLAN.

If there are ports configured into the dual-mode (default VLAN) state, they can be moved from that state by removing untagged ports the default VLAN that also exist as tagged in a non-default VLAN or removing tagged ports from the non-default VLANs.

Disabling dual-mode for tagged ports

Unless you have a specific need to operate a tagged port in dual-mode, you should make it strictly tagged by removing it from the set of untagged ports in the default VLAN.

If you leave a tagged port in dual-mode as an untagged member of the default VLAN, then any untagged broadcast, multicast, and unknown unicast frames received on that port will be flooded out on all other ports in the default VLAN. This is expected behavior. The **no dual-mode-default-vlan** command has been provided to change this behavior, but this command can only be added if no ports are in dual-mode. If you already have tagged ports and if you do not want them to forward untagged frames, you should remove them from the untagged ports of the default VLAN as shown in this example.

```
device(config)#vlan 2
device(config-vlan-2)#tagged e 1/1 to 1/8
device(config-vlan-2)#vlan 1
device(config-vlan-1)#no untagged e 1/1 to 1/8
```

After you add a port to a VLAN as a tagged member, you should then make it strictly tagged by removing it from the default VLAN as an untagged member.

> NOTE
> If the device already has tagged ports, Brocade strongly recommends that you disable dual-mode for tagged ports by removing all the tagged ports from the set of untagged ports in the default VLAN. Unless the device already has "no dual-mode-default-vlan" configured, or if you intend to use the default VLAN for Layer2 switching of traffic, or if Layer2 switching in the default VLAN is explicitly required for other functions, or if you have your tagged ports configured for dual-mode with untagged traffic from a non-default VLAN.

Layer 2 control protocols on VLANs

Layer 2 protocols such as STP, RSTP, ERP, Foundry MRP, and VSRP can be enabled on a port-based VLAN.

The Layer 2 state associated with a VLAN and port is determined by the Layer 2 control protocol. Layer 2 broadcasts associated with the VLAN will not be forwarded on this port if the Layer 2 state is not FORWARDING.

It is possible that the control protocol, for example STP, will block one or more ports in a protocol-based VLAN that uses a virtual routing interface to route to other VLANs. For IP protocol and IP subnet VLANs, even though some of the physical ports of the virtual routing interface are blocked, the virtual routing interface can still route as long as at least one port in the virtual routing interface's protocol-based VLAN is not blocked by STP.

You can also enable Single STP (SSTP) on the device; however, the ports in all VLANs on which SSTP is enabled become members of a single spanning tree. The ports in VLANs on which SSTP is disabled are excluded from the single spanning tree. A VLAN can also be selectively added or removed from the single spanning tree domain.

Virtual interfaces and CPU protection co-existence on VLANs

CPU protection can be configured on VLANs regardless of whether there are virtual-interfaces configured on them (Previously, CPU protection was only configurable if a virtual-interface was not configured on the VLAN).

There is a difference in the behavior of CPU protection in each of the following situations:

- When virtual-interfaces are configured on a VLAN, the CPU-protection is done only on unknown-unicast packets from the VLAN. Multicast and broadcast packets from the VLAN will be sent to the CPU. This allows the CPU to process packets such as ARP and OSPF "hello" packets that may be relevant to the device.
- When virtual-interface is not configured on the VLAN, the CPU-protection is performed for all packets (unknown-unicast, multicast and broadcast) from the CPU.

Configuring port-based VLANs

As explained above, you can place ports into VLANs to segment traffic into broadcast domains. When you create a VLAN, you specify if ports added to that VLAN are tagged or untagged.

> NOTE
> When adding a port to a VLAN you might get an error message concerning IP routing or IPv6 routing information on the port. If you receive this message, check to see if the port was previously configured for routing protocols such as OSPFv2 or OSPFv3 where the routing protocol was removed globally without first being de-configured on that port. If this is the case, re-enable the routing protocol globally to view the interface configuration and then disable the routing protocol from the port. You can then add the port to the VLAN.

To create a VLAN, perform the tasks listed below.

1. At the global CONFIG level assign an ID to the VLAN.

   ```
   device(config)# vlan 2
   ```

   VLAN IDs can be in the range of 1 - 4090. Use the **no** form of the command to delete the VLAN from the configuration.

   The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

   In addition to a VLAN number, you can assign a name to a VLAN by entering name *vlan-name*. Enter up to 31 characters for name.

2. Once a VLAN ID is assigned, the CLI directs you to the VLAN configuration level. At this level, you add ports to that VLAN and specify if the ports are tagged or untagged.

   ```
   device(config-vlan-2)# untag e 1/9 to 1/16
   device(config-vlan-2)# tagged e 1/1 to 1/8
   ```

   The example above configures a port-based VLAN, VLAN 2. It adds Ethernet ports 1/9 through 1/16 as untagged ports and ports 1/1 through 1/8 as tagged ports. Since ports 1/9 through 1/16 are untagged, they can be members of VLAN 2 only, while ports 1/1 through 1/8 are tagged ports and can be members of other VLANs.

   > NOTE
   > In the configuration above, ports 1/9 - 1/16 are automatically removed from the default VLAN since they are configured as untagged ports; while port 1/1 - 1/8 are still members of the default VLAN.

Syntax: [no] untagged tagged | ethernet slot-number/port-number [to slot-number/port-number | ethernet slot-number/port-number]

Ports are removed from the default VLAN only when the port is added as an **untagged** member of a different VLAN. The **untag** command also allows the ports to process packets that do not contain 802.1q tagging. A port is removed from a default-VLAN when the port is added as an untagged member of a different VLAN.

The **tagged** parameter allows the Brocade device to add a four-byte tag 802.1q tag to the packets that go through the tagged ports. It also allows the ports to be members of other VLANs.

Enter the port that you want to assign to the VLAN for the **ethernet***slot-number* / *port-number* parameter. When you add the LAG group's primary port, all the ports on the LAG group become members of the VLAN.

Use the **no** form of the command to remove the ports from a VLAN.

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

Strictly or explicitly tagging a port

If you want a port to be strictly or explicitly tagged, that port has to be removed from the default VLAN. Enter a command such as the following.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/8
device(config-vlan-2)# vlan 1
device(config-vlan-1)# no untagged e 1/1 to 1/8
```

Assigning or changing a VLAN priority

NOTE
When you apply the vlan priority command with running traffic, it may drop packets for a short period of time and flush out the MAC addresses. This is normal behavior.

You can prioritize traffic on a VLAN by assigning a priority to a VLAN. All packets associated with the VLAN will be classified to the configured priority.

```
device(config-vlan-2)# priority 2
```

Syntax: [no] priority num

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

Assigning a different ID to the default VLAN

As stated above, by default, all ports on a Brocade device belong to the default VLAN, which is VLAN 1, until it is assigned to a port-based VLAN. The default VLAN port membership is always untagged; however, if you want to use VLAN ID 1 as a configurable VLANs with tagged port members, you can assign a different VLAN ID as the default VLAN. Enter commands such as the following command.

```
device(config)# default-vlan-id 4000
```

Syntax: [no] default-vlan-id vlan-id

You must specify a VLAN ID that is not already in use. For example, if VLAN 10 exists, do not use "10" as the new VLAN ID for the default VLAN. VLAN IDs are from 1 - 4090. The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

Configuring protocol-based VLANs

Once port-based VLANs are created, you can further segment the broadcast domains by creating protocol-based VLANs, based on Layer 3 protocols. Use the general procedure below for creating protocol-based VLANs.

1. Create the port-based VLAN that contains the interface that you want to segment using Layer 3 protocols.

```
device(config)# vlan 2
device(config-vlan-2)# untag e 1/9 to 1/16
device(config-vlan-2)# tagged e 1/1 to 1/8
```

2. Under the VLAN configuration level, define the Layer 3 protocol you want to use to segment packets that go through the ports assigned to the port-based VLAN.

```
device(config-vlan-2)# ipv6-proto name Blue
```

   Syntax: [no] ip-proto ipv6-proto | | ipx-proto | atalk-proto | other-proto name protocol-vlan-name

   Enter:

   - – **ip-proto** to create a IP protocol VLAN.
   - – **ipv6-proto** to create a IPv6 protocol VLAN.
   - – **ipx-proto** to create a IPX protocol VLAN.
   - – **atalk-proto** to create an Appletalk protocol VLAN.
   - – **other-proto** to create a protocol VLAN for protocols other than an IP protocol, IPv6, IPX, or Appletalk protocol.

   Enter **name** *vlan-name* if you want to assign a name to the protocol-based VLAN. Enter up to 32 characters for name.

   Use the **no** form of the command to remove the protocol-based VLAN.

3. Assign or exclude specific ports to the protocol-based VLAN.

```
device(config-vlan-group-ipv6-proto)# static e 1/1 e 1/24
device(config-vlan-group-ipv6-proto)# exclude e 1/2 to 1/4
```

   Syntax: [no] static exclude | ethernet slot-number/port-number [to slot-number/port-number]

   The **static** ethernet *slot-number* / *port-number* [to *slot-number* / *port-number*] parameter adds the specified ports within the port-based VLAN as static ports to the protocol-based VLAN. Packets of the specified protocol will be forwarded on these ports.

   The **exclude** ethernet *slot-number* / *port-number* [to *slot-number* / *port-number*] parameter excludes the specified ports from the protocol-based VLAN. Packets of the specified protocol will be dropped if received on these ports.

Configuring virtual routing interfaces

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

A *virtual routing interface* is a logical routing interface that the Brocade device uses to route Layer 3 protocol traffic between protocol-based VLANs. It is a logical port on which you can configure Layer 3 routing parameters.

For example, to enable a Brocade device to route IP traffic from one IP protocol VLAN to another, you must configure a virtual routing interface on each IP protocol VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/2
```

```
device(
config-vlan-2)# router-interface ve 1
```

The Brocade device can locally route IP packets between VLANs that are defined within a single device.

If you do not need to further partition the port-based VLAN into protocol-based VLANs, you can define a single virtual routing interface at the port-based VLAN level and enable routing on a single virtual routing interface.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/2
device(config-vlan-2)# router-interface ve 2
device(config-vlan-2)# exit
device(config)# interface ve 2
device(config-
ve-2)# ip address 10.1.1.1/24
```

Syntax: router-interface ve ve-number

Enter 1 to the maximum number of virtual routing interfaces supported on the device for *ve-number*.

Integrated Switch Routing

Integrated Switch Routing (ISR) feature enables VLANs configured on the Brocade device to route Layer 3 traffic from one protocol-based VLAN to another instead of forwarding the traffic to an external router. The VLANs provide Layer 3 broadcast domains for the protocols, but do not in themselves provide routing services. This is true even if the source and destination protocols are on the same device.

ISR eliminates the need for an external router by allowing you to route between VLANs using virtual routing interfaces (ves). You configure a separate virtual routing interface on each VLAN that you want to use to route packets. For example, if you configure two IP protocol VLANs on a Brocade device, you can configure a virtual routing interface on each of the IP protocol VLAN, then configure IP routing parameters for the IP protocol VLAN. Thus, the Brocade device forwards IP broadcasts within each VLAN at Layer 2 but routes Layer 3 traffic between the VLANs using the virtual routing interfaces.

> NOTE
> The Brocade device uses the lowest MAC address on the device (the MAC address of port 1/1) as the MAC address for all ports within all virtual routing interfaces you configure on the device.

The routing parameters and the syntax for configuring them are the same as when you configure a physical interface for routing (for example, **interface ve 10**). The logical interface allows the Brocade device to internally route traffic between the protocol-based VLANs without using physical interfaces.

All the ports within a protocol-based VLAN must be in the same port-based VLAN. The protocol-based VLAN cannot have ports in multiple port-based VLANs, unless the ports in the port-based VLAN to which you add the protocol-based VLAN are 802.1q tagged.

You can configure multiple protocol-based VLANs within the same port-based VLAN. In addition, a port within a port-based VLAN can belong to multiple protocol-based VLANs of the same type or different types. For example, if you have a port-based VLAN that contains ports 1/1 - 1/10, you can configure port 1/5 as a member of an AppleTalk protocol VLAN, an IP protocol VLAN, and an IPX protocol VLAN, and so on.

If the router interface for IP is configured on physical ports, then routing occurs independent of the Spanning Tree Protocol (STP). However, if the router interfaces are defined for IP VLAN, they are virtual routing interfaces and are subject to the rules of STP.

If your backbone consists of virtual routing interfaces all within the same STP domain, it is a bridged backbone, not a routed one. This means that the set of backbone interfaces that are blocked by STP will be blocked for routed protocols as well. The routed protocols will be able to cross these paths only when the STP state of the link is FORWARDING. This problem is easily avoided by proper network design.

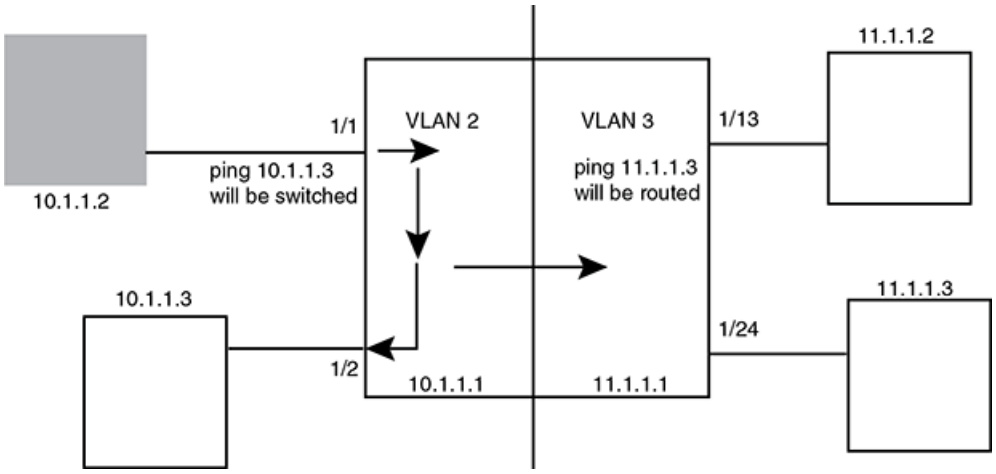
When designing an ISR network, pay attention to your use of virtual routing interfaces and the spanning-tree domain. If Layer 2 switching of your routed protocols (IP, IPX, AppleTalk) is not required across the backbone, then the use of virtual routing interfaces can be limited to edge switch ports within each router. Full backbone routing can be achieved by configuring routing on each physical interface that connects to the backbone. Routing is independent of STP when configured on a physical interface.

If your ISR design requires that you switch IP, IPX, or Appletalk at Layer 2 while simultaneously routing the IP protocol over a single backbone, then create multiple port-based VLANs and use VLAN tagging on the backbone links to separate your Layer 2 switched and Layer 3 routed networks.

There is a separate STP domain for each port-based VLAN. Routing occurs independently across port-based VLANs or STP domains. You can define each end of each backbone link as a separate tagged port-based VLAN. Routing will occur independently across the port-based VLANs. Because each port-based VLAN's STP domain is a single point-to-point backbone connection, you are guaranteed to never have an STP loop. STP will never block the virtual router interfaces within the tagged port-based VLAN, and you will have a fully routed backbone.

A Brocade device offers the ability to create a virtual routing interface within a Layer 2 STP port-based VLAN or within each IP protocol VLAN. This combination of multiple Layer 2 or Layer 3 broadcast domains and virtual routing interfaces are the basis for the very powerful Integrated Switch Routing (ISR) technology. ISR is very flexible and can solve many networking problems.

FIGURE 10 Example of two separate backbones for the same protocol



The following is a sample configuration for the illustration above.

```
device(config)# vlan 2
device(config-vlan-2)# tagged e 1/1 to 1/2
device(config-vlan-2)# router-inter ve 2
device(config-vlan-2)# exit
device(config)# vlan 3
device(config-vlan-3)# tagged e 1/13 to 1/24
device(config-vlan-3)# router-int ve 3
device(config-vlan-3)# exit
device(config)# interface ve 2
device(config-
ve-2)# ip address 10.1.1.1/24
device(config-if-e1000-2/1)# exit
device(config)# interface ve 3
device(config-
ve-3)# ip address 10.2.1.1/24
```

IP packets are bridged (switched) within the same protocol VLAN if they are on the same subnet; they are routed if they are on a different VLAN.

VLAN groups

To simplify VLAN configuration when you have many VLANs with the same configuration, you can configure *VLAN groups* . When you create a VLAN group, the VLAN parameters you configure for the group apply to all the VLANs within the group.

The VLAN group feature allows you to create multiple port-based VLANs with identical port members. Since the member ports are shared by all the VLANs within the group, you must add the ports as tagged ports. This feature not only simplifies VLAN configuration but also allows you to have a large number of identically configured VLANs in a startup configuration file on the device's flash memory module. Normally, a startup configuration file with a large number of VLANs might not fit on the flash memory module. By grouping the identically configured VLANs, you can conserve space in the startup configuration file so that it fits on the flash memory module.

On the Brocade devices, you can create up to 128 VLAN groups per system.

> NOTE
> Depending on the size of the VLAN ID range you want to use for the VLAN group, you might need to allocate additional memory for VLANs. To allocate additional memory, refer to Allocating memory for more VLANs or virtual routing interfaces on page 196.

Configuring a VLAN group

To configure a VLAN group, perform the tasks listed below.

1. Create the VLAN group and assign the VLANs to that group.

```
device(config)# vlan-group 1 vlan 2 to 1000
```

Syntax: [no] vlan-group *num* **vlan** *vlan-id* **to** *vlan-id*

The *num* parameter specifies the VLAN group ID. On the Brocade devices, you can create up to 128 VLAN groups per system.

The **vlan** *vlan-id* **to** *vlan-id* parameters specify a continuous range (with no gaps) of VLAN IDs that have not been configured in the CLI. Specify the low VLAN ID first and the high VLAN ID second. The command adds all the VLANs in the range to the VLAN group.

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

> NOTE
> The device's memory must be configured to contain at least the number of VLANs you specify for the higher end of the range. For example, if you specify 2048 as the VLAN ID at the high end of the range, you first must increase the memory allocation for VLANs to 2048 or higher. Refer to Allocating memory for more VLANs or virtual routing interfaces on page 196.

2. The CLI directs you to the VLAN group configuration level. Add tagged ports to the group. Since all the VLANs in the group share the ports, you must add the ports as tagged ports.

```
device(config-vlan-group-1)# tagged e 1/1 to 1/2
```

Syntax: [no] tagged ethernet [**to** *slot-number/port-number* | **ethernet** *slot-number/port-number*]

Using the **no tagged ethernet** command causes the following error message such as the following to appear.

```
device(config-vlan-10)#no tagged ethernet 4/2
error - ports ether 4/1 to 4/2 are not tagged members of vlan 10
```

This message is normal and indicates that the configuration has take effect. It does not indicate that an error condition has occurred.

3. If required, you can add and remove individual VLANs or VLAN ranges from the VLAN group configuration level. For example, to add VLANs 1001 and 1002 to VLAN group 1 and remove VLANs 900 through 1000, enter the following commands.

```
device(config-vlan-group-1)# add-vlan 1001 to 1002
device(config-vlan-group-1)# remove-vlan 900 to 1000
```

Syntax: [no] add-vlan vlan-id [**to vlan-id**]

Syntax: [no] remove-vlan vlan-id [**to vlan-id**]

Verifying VLAN group configuration

To verify configuration of VLAN groups, display the running configuration file. If you have saved the configuration to the startup configuration file, you also can verify the configuration by displaying the startup configuration file. The following example shows the running configuration information for the VLAN group configured in the previous examples. The information appears in the same way in the startup configuration file.

```
device(config)# show running-config
```

lines not related to the VLAN group omitted...

```
vlan-group 1 vlan 2 to 900
add-vlan 1001 to 1002
tagged ethernet 1/1 to 1/2
```

Displaying information about VLAN groups

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

```
device# show vlan-group 10
Configured VLAN-Group entries : 1
Maximum VLAN-Group entries : 32
VLAN-GROUP 10
Number of VLANs: 4
VLANs: 10 to 13
Tagged ports: ethernet 3/1
```

The example shows configuration information for two VLAN groups, group 1 and group 2.

Syntax: show vlan-group [**group-id**]

The *group-id* specifies a VLAN group. If you do not use this parameter, the configuration information for all the configured VLAN groups is displayed.

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 Brocade device is deployed in a Metro network and provides forwarding for two Foundry MRP rings that each contain 128 VLANs, you can configure a topology group for each ring. If a link failure in a ring causes a topology change, the change is applied to all the VLANs in the ring's topology group. Without topology groups, you would need to configure a separate ring for each VLAN.

You can use topology groups with the following Layer 2 protocols:

- STP
- Foundry MRP
- VSRP
- RSTP
- Ethernet RIng Protection (ERP)

Master VLAN and member VLANs

Each topology group contains a master VLAN and can contain one or more member VLANs and VLAN groups. A definition for each of these VLAN types follows:

- **Master VLAN** - The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for Foundry MRP, the topology group's master VLAN contains the ring configuration information.
- **Member VLANs** - The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the member VLANs. Member VLANs do not independently run a Layer 2 protocol. VPLS VLANs can become member VLANs within a topology group.
- **Member VLAN groups** - A VLAN group is a named set of VLANs. The VLANs within a VLAN group have the same ports and use the same values for other VLAN parameters.

When a Layer 2 topology change occurs on a port in the master VLAN, the same change is applied to that port in all the member VLANs that contain the port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Master VLANs and customer VLANs in Foundry MRP

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as Foundry MRP. For more information on topology group and Foundry MRP, refer to the *VLANs* chapter.

Control ports and free ports

A port in a topology group can be a control port or a free port:

- A **Control port** is a port in the master VLAN and, therefore, is controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs is controlled by the master VLAN's Layer 2 protocol. Each member VLAN must contain all of the control ports. All other ports in the member VLAN are "free ports."
- **Free ports** are not controlled by the master VLAN's Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs that are not also in the master VLAN are free ports.

  NOTE
  Because free ports are not controlled by the master port's Layer 2 protocol, they are assumed always to be in the forwarding state, when enabled.

Configuration considerations

The configuration considerations are as follows:

- You can configure up to 255 topology groups. Each group can control up to 4000 VLANs. A VLAN cannot be controlled by more than one topology group.
- The topology group must contain a master VLAN and can also contain individual member VLANs, VLAN groups, or a combination of individual member VLANs and VLAN groups. Therefore, you configure the master VLAN and member VLANs or member VLAN groups before you configure a topology group.
- After you add a VLAN as a member of a topology group, the device deletes all the Layer 2 protocol information on that VLAN.
- If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.
- If you remove the master VLAN (by entering the **no master-vlan** command with the *vlan-id* variable), the software selects the new master VLAN from member VLANs. A new candidate master VLAN will be in configured order to a member VLAN so that the first added member VLAN will be a new candidate master VLAN. Once you save and reload, a member VLAN with the youngest VLAN ID will be a new candidate master. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.
- After you add a VLAN or VLAN group as a member of a topology group, all the Layer 2 protocol configuration information for the VLAN or group is deleted. For example, if STP is configured on a VLAN and you add the VLAN to a topology group, the STP configuration is removed from the VLAN. After you add the VLAN to a topology group, the VLAN uses the Layer 2 protocol settings of the master VLAN. If you remove a member VLAN or VLAN group from a topology group, you need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.
- On platforms where the Ethernet Service Instance (ESI) framework is supported, master VLANs in a topology group must either be in the default ESI or within the same ESI. Master and member VLANs cannot span multiple ESIs.

Configuring a topology group

To configure a topology group, enter commands such as the following.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan 2
device(config-topo-group-2)# member-vlan 3
device(config-topo-group-2)# member-vlan 4
device(config-topo-group-2)# member-vlan 5
device(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 **topology-group** command creates a topology group. The *group-id* parameter assigns an ID 1 to 255 to the topology group.

Syntax: [no] master-vlan vlan-id

This command adds the master VLAN to the topology group. The VLAN must already be configured. Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

NOTE
When a port is added to a master VLAN, it will be added as a free port. Similarly when a port has to be removed from master VLAN, first disable any the Layer 2 protocol on the port, then remove the port from the master VLAN.

Syntax: [no] member-vlan vlan-id

This command adds a member VLAN to the topology group. The VLAN must already be configured.

Syntax: [no] member-group num

This command adds a VLAN group to the topology group. The *num* specifies a VLAN group ID. The VLAN group must already be configured.

Adding VPLS VLANs to topology groups

To add *single-tagged or untagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following example.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan 2
device(config-topo-group-2)# member-vlan vpls id 34 vlan 42 to 45
```

To add *dual-tagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following configuration example.

```
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 10
device(config-topo-group-1)# member-vlan 20
device(config-topo-group-1)# member-vlan vpls id 5 vlan 300 inner-vlan 20 to 25
```

Syntax: [no] member-vlan vpls [**id** *vpls-id* | **name** *vpls-name* } **vlan** *vlan-id* [**to** *vlan-id*]

OR

Syntax: [no] member-vlan vpls [**id** *vpls-id* | **name** *vpls-name*] **vlan** *vlan-id* [**inner-vlan** *inner-vlan-id* [**to** *inner-vlan-id*]]

The **id** option allows you to specify the VPLS instance that you are configuring into the topology group by using the VPLS ID of the instance. A value in the range of 1 - 4294967294 can be entered for VPLS ID.

The **name** option allows you to specify the VPLS instance that you are configuring into the topology group by using the name of the instance.

The *vlan-id* variable is used with the **vlan** keyword to specify the VPLS VLAN being configured into topology group. You can specify multiple *vlan-id* values or specify a range of VLANs using the **to** option.

The **inner-vlan** option allows you to specify a VPLS dual-tagged (double-tagged) VLAN configuration.

NOTE
The **inner-vlan** option does not allow both outer VLAN ranges and inner VLAN ranges for a given VPLS instance. Once an outer VLAN range is specified, the inner VLAN option is not allowed. However, if a single outer VLAN is specified, the inner VLAN option and range is allowed.

NOTE
You cannot delete a topology master VLAN if the topology group has only VPLS VLAN members and no Layer 2 VLAN members because the normal procedure for deleting a topology master VLAN is to elect another Layer 2 VLAN as the new master. Because a VPLS VLAN cannot be a master VLAN, you must have at least one Layer 2 VLAN as a member. If it does not currently exist, you must add a Layer 2 VLAN before deleting a topology master.

NOTE
A maximum of 4000 VPLS member VLANs can be added to a topology group.

Topology group support within an ESI

Topology groups can be configured with VLANs that are part of a user-defined ESI. (Consult Topology Groups on page 177 to see which platform supports topology groups within an ESI.) When you configure topology groups in such a scenario, both the master and member VLANs must be part of the same ESI. If an ESI is not specified, the system assumes a reference to the default ESI. Below is an example of configuring topology groups with VLANs that are part of a user-defined ESI.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan service-esi 2
device(config-topo-group-2)# member-vlan service-esi 3
device(config-topo-group-2)# member-vlan service-esi 4
device(config-topo-group-2)# member-vlan service-esi 5
device(config-topo-group-2)# member-group service-esi 2
```

The commands configure topology group 2 and add the following to it:

- VLAN 2 in ESI "service-esi" as master VLAN
- VLANs 3, 4, and 5 in ESI "service-ESI" as member VLANs
- Member VLAN group 2

Syntax: [no] topology-group group-id

This command creates a topology group. The *group-id* parameter assigns an ID in the range 1 to 255 to the topology group.

Syntax: [no] master-vlan esi-name vlan-id

This command adds the master VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name". Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

Syntax: [no] member-vlan esi-name vlan-id

This command adds a member VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name".

Syntax: [no] member-group esi-name num

This command adds a VLAN group in ESI identified by "esi-name" to the topology group. The *num* specifies a VLAN group ID. The VLAN group must already be configured.

Displaying topology group information

This section contains examples of the **show topology-group** command output. Support for topology groups within an ESI is supported on a minority of platforms (listed in Displaying topology group information on a Brocade NetIron CES Series device on page 183), so its example appears at the end of this section.

Displaying topology group information on a Brocade NetIron XMR Series or Brocade NetIron MLX Series device

The **show topology-group** command offers a choice between one of two mandatory parameters. The command syntax (on a Brocade NetIron XMR Series or Brocade NetIron MLX Series device) is as follows.

Syntax: show topology-group *group-id* | **hw-index-table** [*hw-index*]

The first example in this section utilizes the first possible mandatory parameter, *group-id*. The second example utilizes the second possible mandatory parameter, **hw-index-table**, along with an optional variable, a hardware index number.

Display topology group information by using a Group ID

To display topology group information for group 10, enter the **show topology-group** command.

```
device#show topology-group 10
Topology Group 10
==================
Topo HW Index   : 0
Master VLAN     : 10
VPLS VLAN exist : TRUE
Member VLAN     : 20
Member Group  : None
Control Ports : ethe 3/11 to 3/12 ethe 3/15 to 3/16
Free Ports :
```

Syntax: show topology-group group-id

This display shows the following information:

TABLE 24 CLI display of topology group information

This field...	Displays...
Topology Group	The ID of the topology group. The range for *group-id* is 1 - 256.
Topo HW Index	A topology hardware index is a unique hardware ID that is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The range for *hw-index* is 0 - 511. (The **show topology-group hw-index-table** command output shows the mapping of a topology hardware index to a VLAN.)
Master-VLAN	The master VLAN for the topology group. The settings for STP, Foundry MRP, ERP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
VPLS VLAN exist	Indicates whether the topology group has one or more VPLS VLANs as a topology group member. The content of this field is TRUE or FALSE.
Member-VLAN	The VLAN ID of the member of the topology group.
Control ports	The master VLAN ports that are configured with Layer 2 protocol information. The Layer 2 protocol configuration and state of these ports in the master VLAN applies to the same port numbers in all the member VLANs.
Free ports	A list of all free ports in the topology group. A free port is not controlled by the Layer 2 protocol information in the master VLAN. In the example screen output, the absence of any number indicates that no ports are free.

Display topology group information by using hardware index table numbers

Display the information for hardware index table 0.

```
device#show topology-group hw-index-table  0
Total Instances : 512
Free Instances  : 511
Topo HW Index    Vlan ID
-------------    -------
0                10
```

Syntax: show topology-group hw-index-table [**hw-index**]

The range for *hw-index* is 0 - 511. If you do not specify a number for *hw-index*, the output screen lists all entries.

TABLE 25 Topology group information with hardware index table

This field...	Displays...
Total Instances	Total number of topology hardware indexes that have been initialized in the system.
Free Instances	Number of free topology hardware indexes that are left in the system.
Topology HW Index	A topology hardware index is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The **show topology-group hw-index-table** command output shows the mapping of a topology hardware index to a VLAN. The range for is 0 - 511. In the example, hardware index table 0 is mapped to the VLAN with an ID of 10.
VLAN ID	The ID of the port-based VLAN that owns the protocol instance on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on a device, all protocol information is for VLAN 1.

Displaying topology group information on a Brocade NetIron CES Series device

To display topology group information within an ESI, enter the **show topology-group** command, as in the following example.

```
device(config)# show topology-group 3
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 26 CLI display of topology group information

This field...	Displays...
master-vlan	The master VLAN for the topology group. The settings for STP, Foundry MRP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
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: • Foundry MRP • STP • RSTP

TABLE 26 CLI display of topology group information (continued)

This field...	Displays...
	• VSRP • ERP
Per vlan free ports	The ports that are not controlled by the Layer 2 protocol information in the master VLAN.

Configuring super aggregated VLANs

A super aggregated VLAN allows multiple VLANs to be placed within another VLAN. This feature allows you to construct Layer 2 paths and channels. A path contains multiple channels, each of which is a dedicated circuit between two end points. The two devices at the end points of the channel appear to each other to be directly attached. The network that connects them is transparent to the two devices.

You can aggregate up to 4090 VLANs within another VLAN. This provides a total VLAN capacity on one Brocade device of 16,728,100 channels (4090 * 4090).

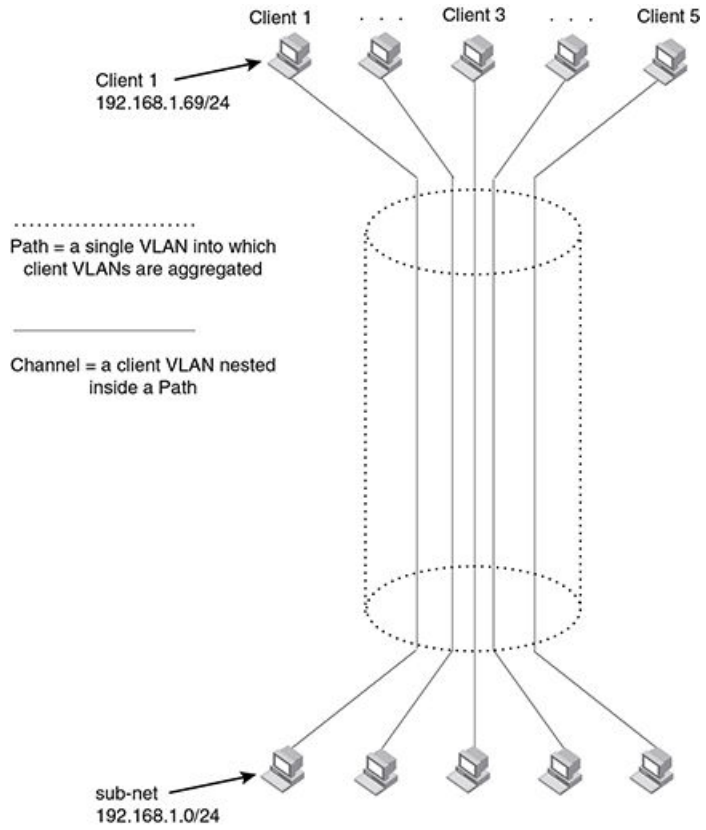
The devices connected through the channel are not visible to devices in other channels. Therefore, each client has a private link to the other side of the channel.

Super aggregated VLANs are useful for applications such as Virtual Private Network (VPN) or Transparent LAN Services (TLS) in which you need to provide a private, dedicated Ethernet connection to individual clients to transparently reach its subnet across multiple networks. The feature allows point-to-point and point-to-multipoint connections.

Figure 11 shows a conceptual picture of the service that aggregated VLANs provide.

In Super Aggregated VLANs, the outer VLAN (path) and the inner VLAN (channel) use different tag types. For example, the outer VLAN tag-type can be 9100 and the inner VLAN tag-type can be 8100 as shown in Figure 11.

FIGURE 11 Conceptual model of the super aggregated VLAN application



Each client connected to the edge device is in its own port-based VLAN. All the clients' VLANs are aggregated by the edge device into a single VLAN for connection to the core.

The device that aggregates the VLANs forwards the aggregated VLAN traffic through the core. The core can consist of multiple devices that forward the aggregated VLAN traffic. The edge device at the other end of the core separates the aggregated VLANs into the individual client VLANs before forwarding the traffic. The edge devices forward the individual client traffic to the clients. For the clients' perspective, the channel is a direct point-to-point link.

Figure 12 shows an example application that uses aggregated VLANs. This configuration includes the client connections shown in Figure 11.

FIGURE 12 Example super aggregated VLAN application

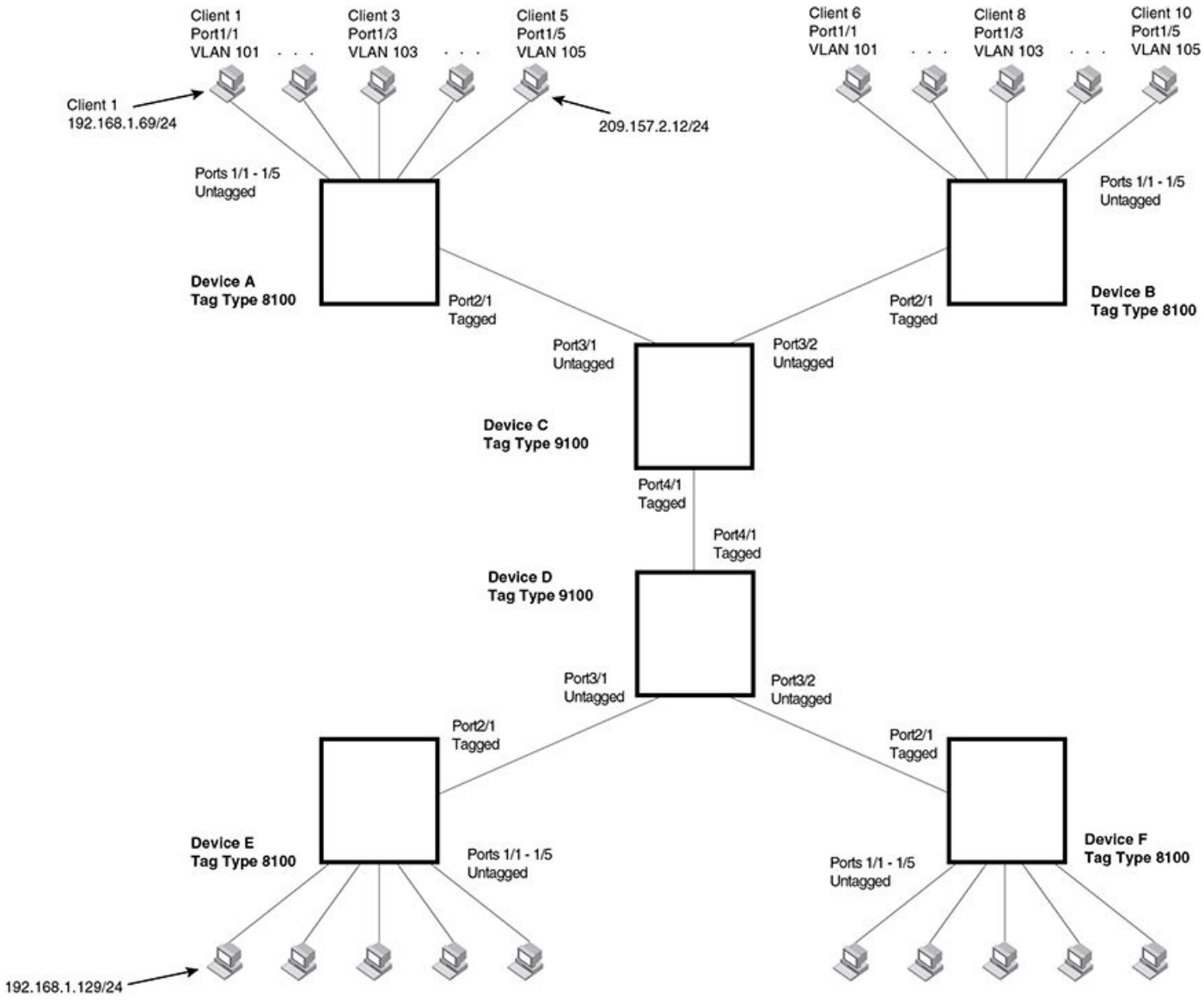


Figure 12 shows a collocation service provides private channels for multiple clients. Although the same devices are used for all the clients, the VLANs ensure that each client receives its own Layer 2 broadcast domain, separate from the broadcast domains of other clients. For example, client 1 cannot ping client 5.

The clients at each end of a channel appear to each other to be directly connected and thus can be on the same subnet and use network services that require connection to the same subnet. In this example, client 1 is in subnet 192.168.1.0/24 and so is the device at the other end of client 1's channel.

Since each VLAN configured on the core devices is an aggregate of multiple client VLANs, the aggregated VLANs greatly increase the number of clients a core device can accommodate.

This example shows a single link between the core devices. However, you can use a LAG group to add link-level redundancy.

Configuring aggregated VLANs

A maximum of 1526 bytes are supported on ports where super-aggregated VLANs are configured. This allows for an additional 8 bytes over the untagged port maximum to allow for support of two VLAN tags.

To configure aggregated VLANs, configure tagged and untagged VLANs on the edge device, then configure the aggregated and other VLANs on the core device. Perform the tasks listed below.

1. On each edge device, configure a separate port-based VLAN for each client connected to the edge device. In each client VLAN:
   - Add the port connected to the client as an untagged port.
   - Add the port connected to the core device (the device that will aggregate the VLANs) as a tagged port. This port must be tagged because all the client VLANs share the port as an uplink to the core device.

   For example, to configure device A in Configuring super aggregated VLANs on page 184, enter commands such as the following.

```
device(config)# vlan 101
device(config-vlan-101)# tagged ethernet 2/1
device(config-vlan-101)# untagged ethernet 1/1
device(config-vlan-101)# exit
device(config)# vlan 102
device(config-vlan-102)# tagged ethernet 2/1
device(config-vlan-102)# untagged ethernet 1/2
device(config-vlan-102)# exit
device(config)# vlan 103
device(config-vlan-103)# tagged ethernet 2/1
device(config-vlan-103)# untagged ethernet 1/3
device(config-vlan-103)# exit
device(config)# vlan 104
device(config-vlan-104)# tagged ethernet 2/1
device(config-vlan-104)# untagged ethernet 1/4
device(config-vlan-104)# exit
device(config)# vlan 105
device(config-vlan-105)# tagged ethernet 2/1
device(config-vlan-105)# untagged ethernet 1/5
device(config-vlan-105)# exit
device(config)# write memory
```

   Syntax: [no] vlan vlan-id

   Syntax: [no] untagged tagged | ethernet slot-number/port-number [to slot-number/port-number | ethernet slot-number/port-number]

   The **tagged** command adds the port that the device uses for the uplink to the core device.

   The **untagged** command adds the ports connected to the individual clients.

2. On each core device:
   - Configure a VLAN tag type (tag ID) that is different than the tag type used on the edge devices. If you use the default tag type (8100) on the edge devices, set the tag type on the core devices to another value, such as 9100. The tag type must be the same on all the core devices. The edge devices also must have the same tag type but the type must be different from the tag type on the core devices.

     NOTE

     You can enable the Spanning Tree Protocol (STP) on the edge devices or the core devices, but not both. If you enable STP on the edge devices and the core devices, STP will prevent client traffic from travelling through the core to the other side.

For example, to configure the aggregated VLANs on device C in Configuring super aggregated VLANs on page 184, enter the following commands.

```
device(config)# tag-type 9100
device(config)# vlan 101
device(config-vlan-101)# tagged ethernet 4/1
device(config-vlan-101)# untagged ethernet 3/1
device(config-vlan-101)# exit
device(config)# vlan 102
device(config-vlan-102)# tagged ethernet 4/1
device(config-vlan-102)# untagged ethernet 3/2
device(config-vlan-102)# exit
device(config)# write memory
```

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

The *num* variable is the hexadecimal ethernet tag type. Default value is 8100.

Complete CLI examples

The following sections show all the Aggregated VLAN configuration commands on the devices in Configuring super aggregated VLANs on page 184.

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 Configuring super aggregated VLANs on page 184 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

```
device-A(config)# vlan 101
device-A(config-vlan-101)# tagged ethernet 2/1
device-A(config-vlan-101)# untagged ethernet 1/1
device-A(config-vlan-101)# exit
device-A(config)# vlan 102
device-A(config-vlan-102)# tagged ethernet 2/1
device-A(config-vlan-102)# untagged ethernet 1/2
device-A(config-vlan-102)# exit
device-A(config)# vlan 103
device-A(config-vlan-103)# tagged ethernet 2/1
device-A(config-vlan-103)# untagged ethernet 1/3
device-A(config-vlan-103)# exit
device-A(config)# vlan 104
device-A(config-vlan-104)# tagged ethernet 2/1
device-A(config-vlan-104)# untagged ethernet 1/4
device-A(config-vlan-104)# exit
device-A(config)# vlan 105
device-A(config-vlan-105)# tagged ethernet 2/1
device-A(config-vlan-105)# untagged ethernet 1/5
device-A(config-vlan-105)# exit
device-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.

```
device-B(config)# vlan 101
device-B(config-vlan-101)# tagged ethernet 2/1
device-B(config-vlan-101)# untagged ethernet 1/1
device-B(config-vlan-101)# exit
device-B(config)# vlan 102
device-B(config-vlan-102)# tagged ethernet 2/1
device-B(config-vlan-102)# untagged ethernet 1/2
device-B(config-vlan-102)# exit
device-B(config)# vlan 103
device-B(config-vlan-103)# tagged ethernet 2/1
device-B(config-vlan-103)# untagged ethernet 1/3
device-B(config-vlan-103)# exit
device-B(config)# vlan 104
device-B(config-vlan-104)# tagged ethernet 2/1
device-B(config-vlan-104)# untagged ethernet 1/4
device-B(config-vlan-104)# exit
device-B(config)# vlan 105
device-B(config-vlan-105)# tagged ethernet 2/1
device-B(config-vlan-105)# untagged ethernet 1/5
device-B(config-vlan-105)# exit
device-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.

```
device-C(config)# tag-type 9100
device-C(config)# vlan 101
device-C(config-vlan-101)# tagged ethernet 4/1
device-C(config-vlan-101)# untagged ethernet 3/1
device-C(config-vlan-101)# exit
device-C(config)# vlan 102
device-C(config-vlan-102)# tagged ethernet 4/1
device-C(config-vlan-102)# untagged ethernet 3/2
device-C(config-vlan-102)# exit
device-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.

```
device-D(config)# tag-type 9100
device-D(config)# vlan 101
device-D(config-vlan-101)# tagged ethernet 4/1
device-D(config-vlan-101)# untagged ethernet 3/1
device-D(config-vlan-101)# exit
device-D(config)# vlan 102
device-D(config-vlan-102)# tagged ethernet 4/1
device-D(config-vlan-102)# untagged ethernet 3/2
device-D(config-vlan-102)# exit
device-D(config)# write memory
```

Commands for device E

Since the configuration in Configuring super aggregated VLANs on page 184 is symmetrical, the commands for configuring device E are identical to the commands for configuring device A.

```
device-E(config)# vlan 101
device-E(config-vlan-101)# tagged ethernet 2/1
device-E(config-vlan-101)# untagged ethernet 1/1
device-E(config-vlan-101)# exit
device-E(config)# vlan 102
device-E(config-vlan-102)# tagged ethernet 2/1
device-E(config-vlan-102)# untagged ethernet 1/2
device-E(config-vlan-102)# exit
device-E(config)# vlan 103
device-E(config-vlan-103)# tagged ethernet 2/1
device-E(config-vlan-103)# untagged ethernet 1/3
device-E(config-vlan-103)# exit
device-E(config)# vlan 104
device-E(config-vlan-104)# tagged ethernet 2/1
device-E(config-vlan-104)# untagged ethernet 1/4
device-E(config-vlan-104)# exit
device-E(config)# vlan 105
device-E(config-vlan-105)# tagged ethernet 2/1
device-E(config-vlan-105)# untagged ethernet 1/5
device-E(config-vlan-105)# exit
device-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 Configuring super aggregated VLANs on page 184 are symmetrical, the configuration of device F is also identical to the configuration of device A and device B.

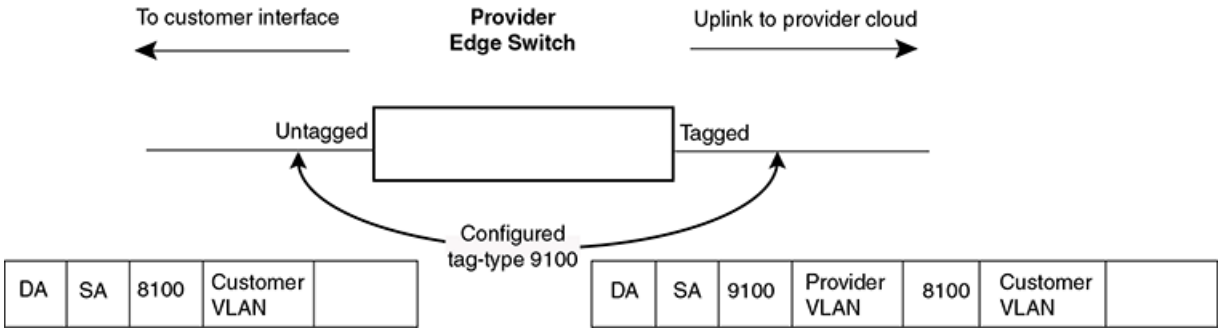
```
device-F(config)# vlan 101
device-F(config-vlan-101)# tagged ethernet 2/1
device-F(config-vlan-101)# untagged ethernet 1/1
device-F(config-vlan-101)# exit
device-F(config)# vlan 102
device-F(config-vlan-102)# tagged ethernet 2/1
device-F(config-vlan-102)# untagged ethernet 1/2
device-F(config-vlan-102)# exit
device-F(config)# vlan 103
device-F(config-vlan-103)# tagged ethernet 2/1
device-F(config-vlan-103)# untagged ethernet 1/3
device-F(config-vlan-103)# exit
device-F(config)# vlan 104
device-F(config-vlan-104)# tagged ethernet 2/1
device-F(config-vlan-104)# untagged ethernet 1/4
device-F(config-vlan-104)# exit
device-F(config)# vlan 105
device-F(config-vlan-105)# tagged ethernet 2/1
device-F(config-vlan-105)# untagged ethernet 1/5
device-F(config-vlan-105)# exit
device-F(config)# write memory
```

Configuring 802.1q-in-q tagging

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

Figure 13 shows an 802.1Q configuration example.

FIGURE 13 SAV configuration example

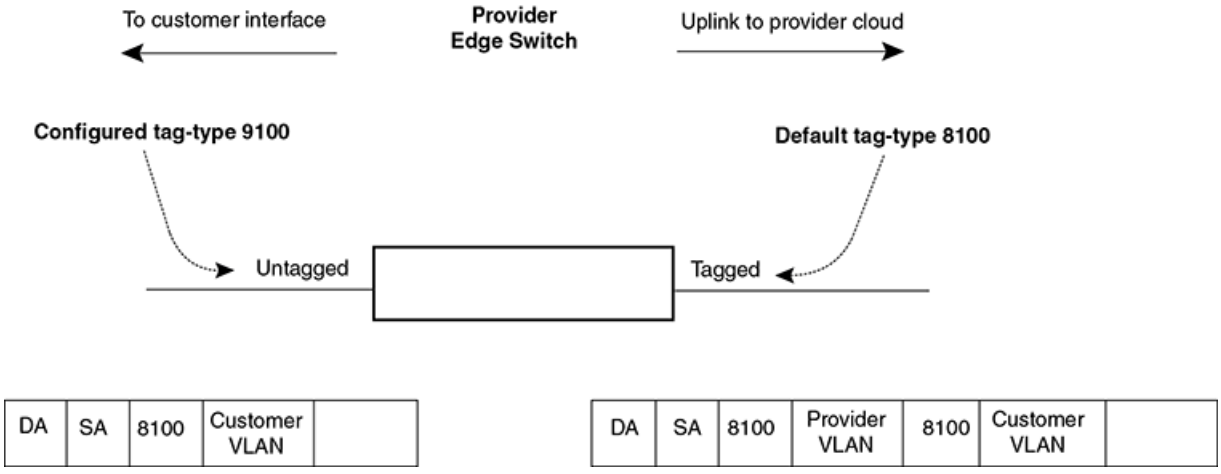


As shown in Figure 13, the ports to customer interfaces are untagged, whereas the uplink ports to the provider cloud are tagged, because multiple client VLANs share the uplink to the provider cloud. In this example, the Brocade device treats the customer's private VLAN ID and 8100 tag type as normal payload, and adds the 9100 tag type to the packet when the packet is sent to the uplink and forwarded along the provider cloud.

As long as the switches in the provider's network support the 9100 tag type, the data gets switched along the network. However, devices that do not support the 9100 tag type may not properly handle the packets.

Figure 14 and Example configuration on page 192 show an example application of 802.1Q-in-Q.

FIGURE 14 802.1Q-in-Q configuration example



In Configuring 802.1q tag-type translation on page 193, 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 Brocade device will switch the frames to the uplink device with an additional 8100 tag, thereby supporting devices that only support this method of VLAN tagging.

Configuration rules

Follow the rules below when configuring 802.1q-in-q tagging:

- The Brocade device supports per port tag-type configuration. Consequently, each port can have its own tag-type setting.
- The default tag-type for a port is 8100.
- The Brocade device supports 802.1q-in-q tagging where the inner and outer tag can have different or same tag-type values. This feature maximizes interoperability with third-party devices.

Enabling 802.1Q-in-Q tagging

To enable the 802.1Q-in-Q feature, configure an 802.1Q tag type on the untagged edge links (the customer ports) to any value other than the 802.1Q tag for incoming traffic.

For example, in Example configuration on page 192, 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 Example configuration on page 192, enter commands such as the following on the untagged edge links of devices C and D.

```
device(config)# tag-type 9100 e 3/1 to 3/2
```

Syntax: [no] tag-type num [**ethernet slot-number/port-number** [**to slot-number/port-number**]]

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

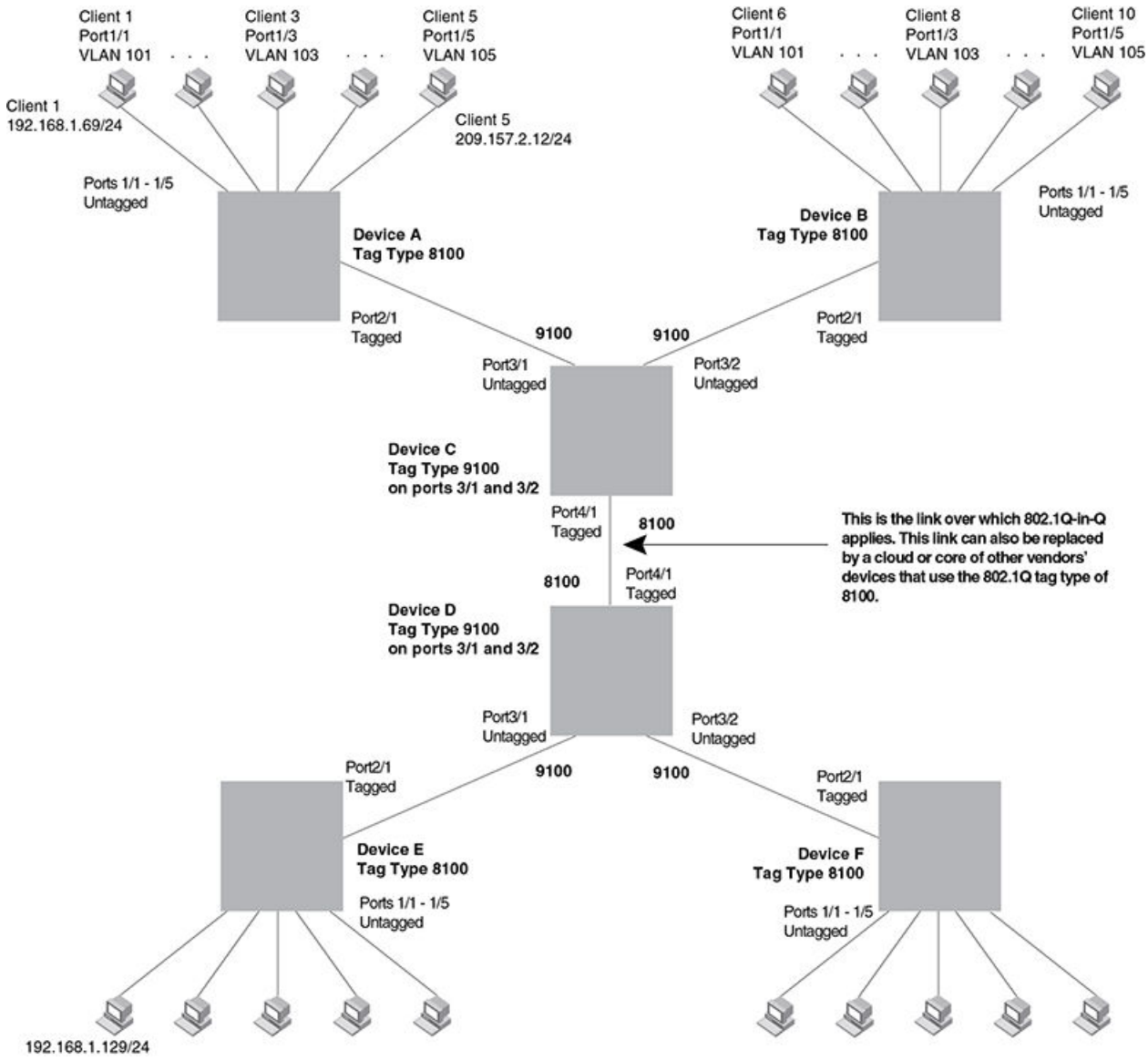
The **ethernet** *port number* **to** *port number* parameter specifies the ports that will use the defined 802.1Q tag. This parameter operates with the following rules:

- If you do not specify a port or range of ports, the 802.1Q tag applies to all Ethernet ports on the device.

Example configuration

Figure 15 shows an example 802.1Q-in-Q configuration.

FIGURE 15 Example 802.1Q-in-Q configuration



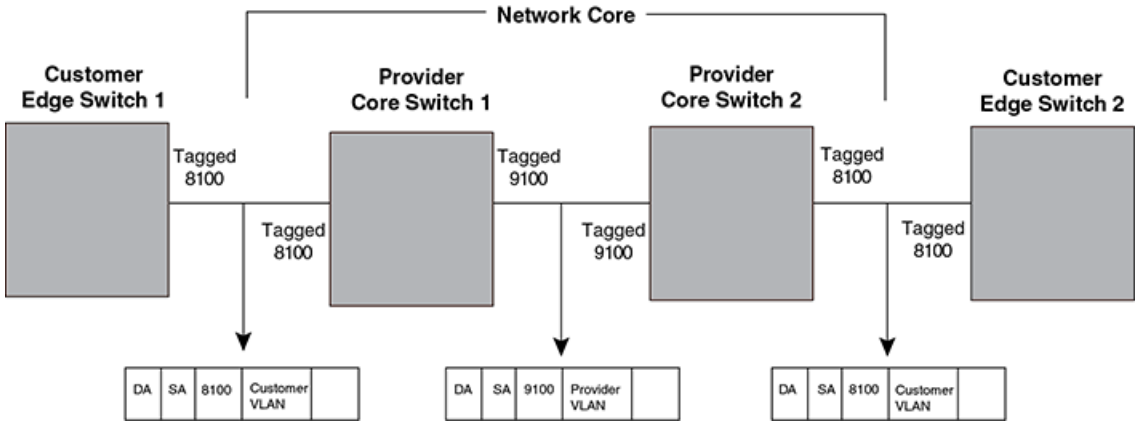
Configuring 802.1q tag-type translation

The introduction of 802.1q tag-type translation provides finer granularity for configuring multiple 802.1q tag-types on a single device, by enabling you to configure 802.1q tag-type per port. This enhancement allows for tag-type translation from one port to the next on tagged interfaces.

802.1Q tag-type translation enables you to configure a separate 802.1q tag-type per port, allowing for tag-type translation from one port to the next on tagged interfaces.

Figure 16 shows a basic example application of the 802.1q tag-type translation feature.

FIGURE 16 802.1q Tag-type translation configuration example 1



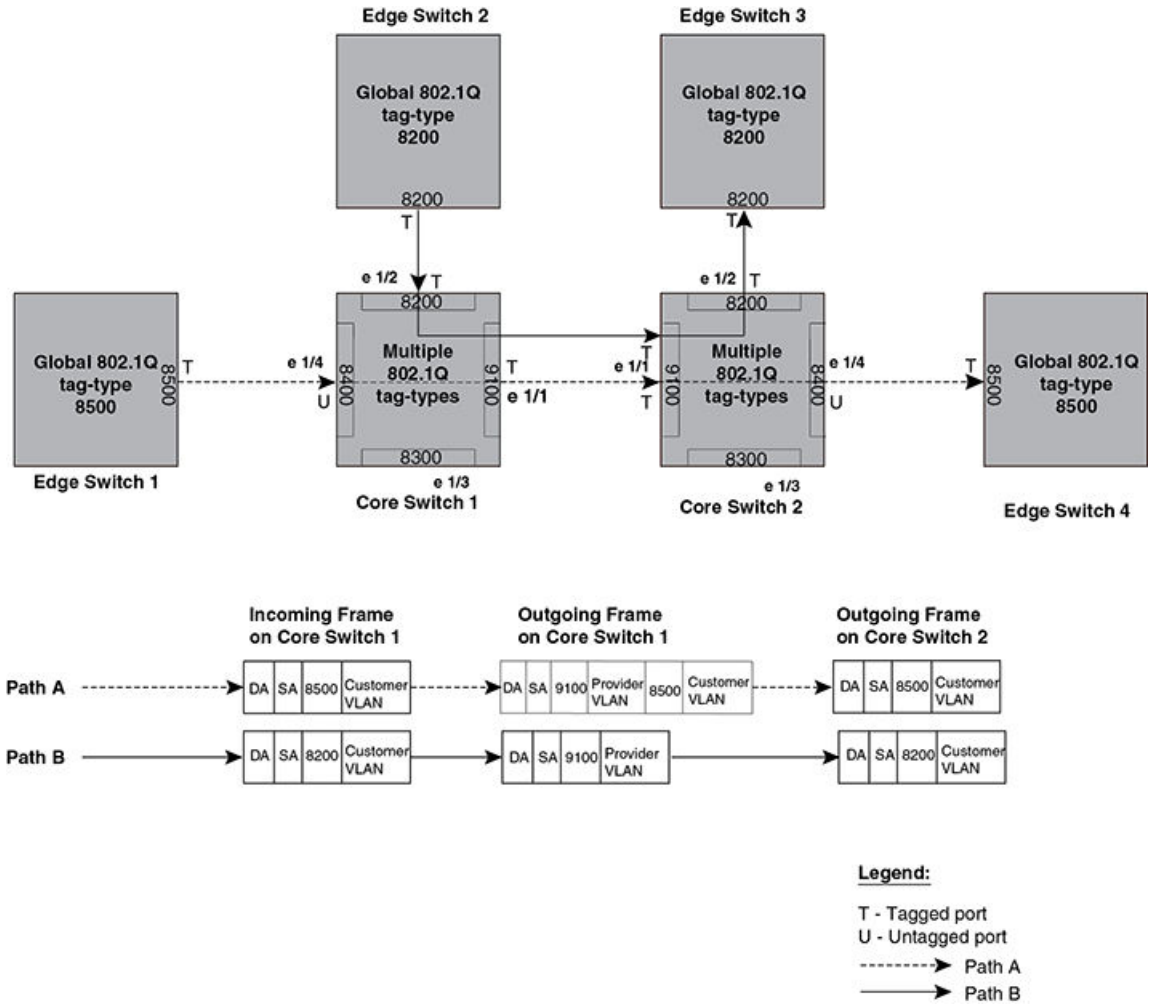
As illustrated in Figure 16, the devices process the packet as follows:

- 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 16 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 17 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 17 802.1q Tag-type translation configuration example 2



As illustrated in Figure 17, the devices process the packets as follows:

- Path A: When Core Switch 1 receives the tagged packet from Edge Switch 1, it keeps the 8500 tag-type in the frame header (because the incoming port on Core Switch 1 is untagged) and adds the 9100 tag-type as it sends the packet to the uplink (Core Switch 2). In this case, the packet is double-tagged as it travels between the core devices.
- Path B: When Core Switch 1 receives the tagged packet from Edge Switch 2, it removes the 8200 tag-type and replaces (translates) it with the 9100 tag-type as it sends the packet to the uplink (Core Switch 2).

Configuration rules

Configuration of tag-type ports on the Brocade device are on a per-port basis and follow the same rules as described in Configuration rules on page 192.

Enabling 802.1q tag-type translation

To enable 802.1q tag-type translation, configure an 802.1q tag-type on the provider core link, between the provider core switches (refer to Configuring 802.1q tag-type translation on page 193). Enter commands such as the following.

```
device(config)# tag-type 9100 e 1/1
device(config)# tag-type 8200 e 1/2
device(config)# tag-type 8300 e 1/3
device(config)# tag-type 8400 e 1/4
```

Syntax: [no] tag-type num [**ethernet slot-number/port-number** [**to slot-number/port-number**]]

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

The *slot-number/ port-number* [**to***slot-number/ port-number*] parameter specifies the ports that will use the defined 802.1q tag-type. This parameter operates with the following rules:

- If the port that you specify is part of a multi-slot LAG, the device automatically applies the 802.1q tag-type to all of the ports that are part of the multi-slot LAG.
- If you do not specify a port or range of ports, the 802.1q tag-type applies to all Ethernet ports on the device.

Miscellaneous VLAN features

Allocating memory for more VLANs or virtual routing interfaces

By default, you can configure up to 512 VLANs and virtual routing interfaces on the router. Although this is the default maximum, the Brocade device can support up to 4090 VLANs and 4090 virtual routing interfaces.

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

If you need to configure more than 512 VLANs, enter commands such as the following at the global CONFIG level of the CLI.

```
device(config)# system-max vlan 2048
device(config)# write memory
device(config)# end
device# reload
```

Syntax: [no] system-max vlan num

The *num* parameter specifies the maximum number of VLANs that can be configured.

Syntax: [no] system-max virtual-interface

The *num* parameter specifies the maximum number of virtual-interfaces that can be configured.

> NOTE
> You must reload the system for the new parameters to take effect.

Configuring uplink ports within a port-based VLAN

You can configure a subset of the ports in a port-based VLAN as uplink ports. When you configure uplink ports in a port-based VLAN, the device sends all broadcast and unknown-unicast traffic from a port in the VLAN to the uplink ports, but not to other ports within the VLAN. Thus, the uplink ports provide tighter broadcast control within the VLAN.

For example, if two ports within a port-based VLAN are Gigabit ports attached to the network and the other ports in the VLAN are 10/100 ports attached to clients, you can configure the two ports attached to the network as uplink ports. In this configuration, broadcast

and unknown-unicast traffic in the VLAN does not go to all ports in the VLAN. The traffic goes only to the uplink ports. The clients on the network do not receive broadcast and unknown-unicast traffic from other ports, including other clients.

To configure a port-based VLAN containing uplink ports, enter commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# untag ethernet 1/1 to 1/20
device(config-vlan-10)# untag ethernet 2/1 to 2/2
device(config-vlan-10)# uplink-switch ethernet 2/1 to 2/2
```

Syntax: [no] uplink-switch ethernet port-number [**to port-number** | **ethernet port-number**]

In this example, ports 1 - 20 on slot 1 and ports 1 - 2 on slot 2 are added to port-based VLAN 10. The two ports on slot 2 are then configured as uplink ports.

Configuring control protocols in VLANs

You can configure the following protocols on a VLAN:

- Foundry MRP (Refer to *Metro Ring Protocol* Chapter.)
- ERP (Refer to *Ethernet Ring Protocol* Chapter.)
- VSRP (Refer to *Virtual Switch Redundancy Protocol (VSRP)* Chapter.)
- STP (Refer to *Configuring Spanning Tree Protocol* Chapter.)
- RSTP (Refer to *Configuring Rapid Spanning Tree Protocol* Chapter.)

Removing tagged or untagged ports

Use the following commands to remove tagged or untagged ports from a VLAN.

Syntax

remove-tagged-ports

remove-untagged-ports

Command Default

This command can only be used when tagged ports and untagged ports have been applied to a VLAN.

Modes

VLAN configuration mode (config-vlan).

Examples

The following example displays the remove-tagged-ports command.

```
device(config-vlan-100)# remove-tagged-ports
Vlan : 100, Ports removed : ethe 1/1 to 1/2 ethe 4/1 to 4/8
device(config-vlan-100)#
```

The following example displays the remove-untagged-ports command.

```
device(config-vlan-100)# remove-untagged-ports
Vlan : 100, Ports removed : ethe 3/1 to 3/24
device(config-vlan-100)#
```

History

Release version	Command history
5.8.00	This command was introduced.

Removing a VLAN

This command removes tagged and untagged ports from all or defined VLANs.

Syntax

remove-vlan { **all** | **vlan** } [**vlan** *vlan_id*][**to** *vlan_id*]

Command Default

This command can only be used when tagged or untagged ports are defined as part of a VLAN.

Parameters

all

Removes all configured VLANs.

vlan

Use with the modifiers to indicate the VLAN range to remove.

vlan_id

Specifies the VLAN where the ports should be removed.

to

Use with the modifiers to indicate the VLAN range to remove.

vlan_id

Specifies the range of VLANs to beremoved.

Modes

User configuration level.

Examples

The following example displays the command with the all option.

```
device(config-if-e100000-1/1)# remove-vlan all
Port ethe 1/1 removed from tagged vlan : 300 400 500 600 700 800 900 1000 2000 3000
4000 and untagged vlan : 200 .
device(config-if-e100000-1/1)#
```

The following example displays the command with a specified VLAN range.

```
device(config-if-e100000-1/2)# remove-vlan vlan 2 to 4090
Port ethe 1/2 removed from tagged vlan : 300 400 500 600 700 800 900 1000 2000 3000
4000 and untagged vlan : 200 .
device(config-if-e100000-1/2)#
```

The following example displays the command that remove a specific VLAN.

```
device(config-if-e10000-4/1)# remove-vlan vlan 500
Vlan : 500, Ports removed : ethe 4/1
device(config-if-e10000-4/1)#
```

History

Release version	Command history
5.8.00	This command was introduced.

Hardware flooding for layer 2 multicast and broadcast packets

Broadcast and multicast packets do not have a specific recipient. In order for these "special" packets to reach their intended recipient, they need to be sent on all ports of the VLAN (or "flooded" across the VLAN).

You must enable hardware flooding for Layer 2 multicast and broadcast packets on the Brocade device. (Layer 2 multicast packets have a multicast address in the destination MAC address field.)

NOTE
This feature is enabled by default on Brocade NetIron CES Series devices.

You can enable hardware flooding for Layer 2 multicast and broadcast packets on a per-VLAN basis.

```
device(config)#
device(config)# vlan 2
device(config-vlan-2)# multicast-flooding
device(config-vlan-2)# exit
```

Syntax: [no] multicast-flooding

NOTE

- This feature cannot be enabled on an empty VLAN; the VLAN must already have ports assigned to it prior to enabling this feature.
- This feature is not supported on Layer 3 protocol-based VLANs.
- If you enable this feature on a VLAN that includes a LAG group, hardware flooding for Layer 2 multicast and broadcast packets occurs only on the LAG group's primary port. Multicast and broadcast traffic for the other ports in the LAG group is handled by software.

Unknown unicast flooding on VLAN ports

Unknown unicast packets do not have a specific (or unicast) recipient. In order for these "special" packets to reach their intended recipient, they needed to be sent on all ports of the VLAN (or "flooded" across the VLAN).

You must enable *hardware* flooding for unknown unicast packets on the Brocade router. It is disabled by default.

NOTE
This feature is enabled by default on the Brocade NetIron CES Series devices.

To enable unicast hardware flooding on a VLAN ports and enable software flooding, enter commands such as the following.

```
device(config)# vlan 2
device(config-vlan-2)# unknown-unicast-flooding
device(config-vlan-2)# exit
```

Syntax: [no] unknown-unicast-flooding

Configuring VLAN CPU protection

VLAN CPU protection is recommended for the VLANs which are intended for pure Layer 2 use. This feature will protect the CPU from the flooding of unknown-unicast or multicast or broadcast Layer 2 packets on that VLAN.

When using routing protocols (such as OSPF and others) on a specific VLAN, you need to disable VLAN CPU protection for it to work. This feature is intended for Layer 2 applications and not for Layer 3 routing applications.

CPU protection can be configured on VLANs regardless of whether there are virtual-interface configured on them (Previously, CPU protection was only configurable if a virtual-interface was not configured on the VLAN).

There is a difference in the behavior of CPU protection in each of the following situations:

- When virtual-interfaces are configured on a VLAN, the CPU-protection is done only on unknown-unicast packets from the VLAN. Multicast and broadcast packets from the VLAN will be sent to the CPU. This allows the CPU to process packets such as ARP and OSPF "hello" packets that may be relevant to the device.
- When virtual-interface is not configured on the VLAN, the CPU-protection is performed for all packets (unknown-unicast, multicast and broadcast) from the CPU.
- With vlan-cpu-protection enabled, currently persistent unknown unicast packets are still sent to the CPU for MAC learning purposes. Although the unknown unicast packets are rate limited to the CPU, it may cause high CPU usage and large CPU Traffic Manager queues, which may cause issues.
- Using the **unknown-unicast-mac-entry** command will forward Layer 2 unknown unicast traffic without going to the CPU.

  NOTE
  This feature is enabled by default on the Brocade NetIron CES Series devices and cannot be disabled.

VLAN CPU protection is enabled per VLAN. To enable VLAN CPU protection on a VLAN, enter the following command.

```
device(config)# vlan 247
device(config-vlan-24)# tagged ethe 4/1 ethe 4/3
device(config-vlan-24)# vlan-cpu-protection
device(config-vlan-24)# unknown-unicast-mac-entry
```

Syntax: [no] vlan-cpu-protection

NOTE
If **vlan-cpu-protection** command is configured for a VLAN, you should not configure **unknown-unicast-flooding** command or **multicast flooding** command on the same VLAN since these features are redundant to **vlan-cpu-protection**.

Syntax: [no] unknown-unicast-mac-entry

NOTE
The **unknown-unicast-mac-entry** command must be configured with the **vlan-cpu-protection** command, as shown in the example above.

Command changes to support Gen-2 modules

The following commands changed to support Gen-2 modules.

Deprecated commands

vlan-counter exclude-overhead

The **vlan-counter exclude-overhead** command has been deprecated in the NetIron XMR/MLX only. The new command is the **exclude-ethernet-overhead** command.

> NOTE
> The **statistics - exclude-ethernet-overhead** command will replace the **vlan-counter exclude-overhead** command when upgrading to the new image.

By default, the VLAN byte counters include the 20-byte Ethernet overhead. You can use the **exclude-ethernet-overhead** command to direct the Brocade device to exclude this overhead when it counts the bytes, as shown in the example below.

```
device(config-statistics)#exclude-ethernet-overhead
```

Syntax: [no] exclude-ethernet-overhead

To disable the configuration, use the **no exclude-ethernet-overhead** command.

> NOTE
> The **vlan-counter exclude-overhead** command is still supported for the Brocade NetIron CES/CER platforms.

byte-accounting

The **byte-accounting** command has been replaced by the **vlan-accounting on|off** command at the VLAN configuration level for the devices.

All Brocade platforms use **vlan-accounting** command at the global (config-vlan-policy) level.

In addition a new global **vlan-policy - vlan-accounting** command has also been introduced to enable/disable accounting for all VLANs.

The **vlan-accounting on | off** command at the VLAN level takes precedence over global configuration. For example, if VLAN accounting is globally enabled, and the user disables VLAN accounting on VLAN 10, then VLAN accounting for VLAN 10 is disabled.

You can configure Brocade NetIron MLX Series devices to count bytes received on a VLAN globally or at the VLAN level. By default, Layer 2 VLAN accounting is globally enabled for all VLANs. The VLAN counters are polled every 50 seconds.

To disable VLAN accounting globally for all VLANs, enter the following command at the config-vlan-policy level of the CLI.

```
device(config-vlan-policy)#no vlan-accounting
```

Syntax: [no] vlan-accounting

To disable VLAN accounting globally, enter the **no vlan-accounting** command.

To configure VLAN accounting for specific VLAN, enter the following command.

```
device(config-vlan-10)# vlan-accounting on
```

Syntax: [no] vlan-accounting on | off

The **vlan-accounting on** command enables counters for a specific VLAN. The **vlan-accounting off** command disables counters for a specific VLAN.

clear vlan all-vlans statistics

The **clear vlan byte-accounting all-vlans** command has been deprecated. The new command is the **clear vlan all-vlans statistics** command.

To clear VLAN counters for all VLANs, enter the following command.

```
device# clear vlan all-vlans statistics
```

Syntax: clear vlan all-vlans statistics

clear vlan byte-accounting

The **clear vlan byte-accounting** command has been has been deprecated. The new command is the **clear vlan statistics** command.

To clear the VLAN counters on a specific VLAN, say VLAN 10, enter the following command.

```
device# clear vlan 10 statistics
```

Syntax: clear vlan *vlan-id* **statistics**

Use the *vlan-id* parameter to specify the name of the VLAN to clear statistic counter on.

Existing display command

The byte counter displayed by the output of **show vlan** command is the number of received bytes across all ports (both G2 and non-G2 ports) in the specified VLAN.

```
device# show vlan
Configured PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 4090
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols : NONE
Untagged Ports : ethernet 2/1 to 2/20 ethernet 3/1 to 3/20 ethernet
PORT-VLAN 2, Name [None], Priority Level0
L2 protocols : NONE
ip-protocol VLAN, Dynamic port disabled
Name: basic
PORT-VLAN 1001, Name [None], Priority Level0
L2 protocols : MRP
Tagged Ports : ethernet 3/1 ethernet 3/12 to 3/13 ethernet 3/20
Bytes received : 6000
```

Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4090	Maximum number of port-based VLANs that you can configure.
Default PORT-VLAN id	ID of the default VLAN.
PORT-VLAN	ID of the port-based VLAN.
Name	Name of the port-based VLAN. [None] appears if a name has not been assigned.
Priority Level	Priority level assigned to the port-based VLAN.
L2 protocols	Layer 2 control protocol configured on the VLAN.
Untagged or Tagged Ports	ID of the untagged or tagged ports that are members of the VLAN.
(protocol-based VLANs)	If protocol based VLANs are configured, their type and name appear after the list of ports.
Bytes received	Displays the number of received bytes across all ports in the specified VLAN.

Extended VLAN counters for 8x10G modules

The NI-MLX-10x8G module supports VLAN counters on the ingress and egress ports. The NI-MLX-10x8G module supports packet and byte accounting for 32K counters on both inbound and outbound traffic on a per-VLAN, per-port, per-priority basis. The 8x10G module supports 64 bit VLAN counters for both packet and byte accounting.

To support extended VLAN accounting, the following modes allow you to configure packet and byte accounting on a 8x10G module.

Priority mode - This mode allows accounting to be performed on a per VLAN, per port, per-priority basis. Priority mode is configured on per-module basis. By default, the per-priority accounting mode is disabled, or in other words, accounting is done per VLAN, per port.

Switched or routed separate mode - This mode allows you to specify whether the switched packet and routed packets should be counted separately or not. This mode is configured globally, and by default, switched packets and routed packets are counted together.

Refer to Table 28 on the number of unique port, VLAN's supported per PPCR based on the configuration of "Priority mode" and "Switched or routed separate mode"..

TABLE 28 Internal priority of switched and routed packets

Switched and routed packets	Account based on the internal priority of the packet- Yes or No	Number of unique port-VLANs that have counters (per-PPCR).
Switch or Route separately	Yes	2047 on ingress and 2047 on egress; each set having 16 counters
Switch or Route separately	No	16383 in ingress and 16383 on egress; each set having 2 counters
Switch or route combined	Yes	4095 on ingress and 4095 on egress; each set having 8 counters
Switch or route combined	No	32767 on ingress and 32767 on egress; each set having 1 counter

Configuring extended VLAN counters

The Gen-2 modules supports the following global configuration commands.

Enabling accounting on per-slot basis

You can enable or disable per-VLAN priority accounting mode on all or a per-slot basis on the ingress and egress counters. To enable accounting on per-slot basis, enter the following command.

Layer 2 VLAN accounting is enabled by default. Counters are polled once every 50 seconds.

```
device(config)#statistics
device(config-statistics)#extended-counters priority all
```

Syntax: [no] extended-counters priority all | *slot-number*

The *slot-number* variable specifies the ID of 8x10 module on which you can perform accounting on per-slot basis.

If the all option is specified, the configuration command is remembered in the system when the 8x10 module is removed.

If you dynamically enable or disable the **extended-counters priority** configuration, the sum of counters displayed on a per-priority basis will not be same as the aggregate count displayed on a per-port or per-VLAN basis.

Enabling accounting on switched or routed packets

To enable or disable accounting on switched packets and routed packets separately, enter the following example:

```
device(config)#statistics
device(config-statistics)#extended-counters routed-switched
```

Syntax: [no] extended-counters routed-switched

If you dynamically enable or disable the **extended-counters routed-switched** configuration, the current counters are saved and added to the count of aggregate packet and byte counters on a per-port or per-VLAN basis and displayed in the output of combined counters.

Displaying VLAN counters

The **show vlan** commands changed to display port-vlan counters for 8x10G modules.

To display VLAN counters information for specific VLAN, enter the following command.

```
device# show vlan 10 statistics
VLAN 10: Extended Routed/Switched Counters (only applicable for G2 modules):
Slot 12: < -- module with per-VLAN/port/priority based accounting
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                           <snip>
     p6   0                0                0                0
     p7   0                0                0                0
eth 12/2  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
                           <snip>
     p6   0                0                0                0
     p7   0                0                0                0
eth 12/3         -- Extended-counter resource allocation failed -   < -- On encountering Stats ID
allocation failure
eth 12/4  0                0                0                0
     p0   0                0                0                0
     p1   0                0                0                0
<snip>
     p6   0                0                0                0
     p7   0                0                0                0
Slot 14: < -- module with per-VLAN/port based accounting
```

Syntax: show vlan *vlan-id* **statistics** [**detail** | **routed** | **switched**]

The *vlan-id* parameter specifies the VLAN ID of the port.

The slot/port parameter specifies the interface module location of a 8x10g module.

Use the **routed** option to view the counters of routed packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **switched** option to view the counters of switched packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **detail** option to view the counters of routed packets, counters of switched packets, and counters of both routed and switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

If the "routed" or "switched" option is specified, counters for only routed and switched packets are displayed respectively, otherwise the output displays combined statistics for both routed and switched packets.

TABLE 29 Output descriptions of the show vlan command

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying VLAN counters for a specific port

To display VLAN counters information for specific port on a VLAN, enter the following command.

```
device# show vlan 10 statistics ethernet 14/1
VLAN 10: Extended Routed/Switched Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display VLAN counters information for routed packets on a specific port on a VLAN, enter the following command.

```
device# show vlan 10 statistics ethernet 14/1 routed
VLAN 10: Extended Routed Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
To display VLAN counters information for switched packets on a specific port on a VLAN, enter the following
command.
device# show vlan 10 statistics ethernet 14/1 switched
VLAN 10: Extended Switched Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1  0                0                0                0
```

To display detailed VLAN counters information on a specific port on a VLAN, enter the following command.

```
device# show vlan 10 statistics ethernet 14/1 detail
VLAN 10: Extended Counters (only applicable for G2 modules):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 14/1
 Routed   0                0                0                0
 Switched 0                0                0                0
 Combined 0                0                0                0
```

Syntax: show vlan *vlanid* **statistics ethernet** *port-id* [**detail** | **routed** | **switched**]

The *vlan-id* parameter specifics the VLAN ID of the port.

The *slot/port* parameter specifies the interface module location of a 8x10g module.

Use the **routed** option to view the counters of routed packets.This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **switched** option to view the counters of switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

Use the **detail** option to view the counters of routed packets, counters of switched packets, and counters of both routed and switched packets. This option is only accepted if the accounting mode is to count routed and switched packets separately.

If the "routed" or "switched" option is specified, counters for only routed and switched packets are displayed respectively, otherwise the output displays combined statistics for both routed and switched packets.

TABLE 30 Output description for the show vlan command

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing extended VLAN counters

You can use the following commands to clear the extended VLAN counters.

Clearing counters for all VLANs

To clear the ingress and egress packet and byte counters for routed packets and switched packets on all VLANs, enter the following command.

```
device# clear vlan all-vlans statistics
```

Syntax: clear vlan all-vlans statistics [**switched**]

Enter the **switched** keyword to clear only the counters of switched packets. If you do not specify the **switched** keyword, the counters for both routed packets and switched packets are cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing counters for a specific VLAN

To clear the VLAN counters for a specific VLAN, enter the following command.

device# clear vlan 10 statistics

Syntax: clear vlan *vlan-id* **statistics** [**switched**]

Use the *vlan-id* option to specify the VLAN ID of the port for which you want to clear the counters.

Enter the **switched** keyword to clear only the counters of switched packets. If you do not specify the **switched** keyword, the counters for both routed packets and switched packets are cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing VLAN and port counters

To clear VLAN, port, and priority counters for specific VLAN and port combinations, enter the following command.

```
device# clear vlan 10 statistics ethernet 1/2 switched
```

Syntax: clear vlan *vlan-id* **statistics ethernet** *port-id* [**switched**]

Use the *vlan-id* option to specify the VLAN ID of the port for which you want to clear the counters.

Use the *port-id* option to specify the port for with you want to clear the counters.

Specify the **switched** keyword to clear counters for the switched packets. If the **switched** keyword is not specified the counters for both routed packets and switched packets will be cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing VLAN counters on a port with a specific priority

To clear counters for a specific port in a VLAN with specific priority, enter the following command.

```
device# clear vlan 10 statistics ethernet 1/2 priority 3 switched
```

Syntax: clear vlan *vlan-id* **statistics ethernet** *port-id* **priority** *0-7* [**switched**]

Use the *vlan-id* option to specify the VLAN ID of the port for which you want to clear the counters.

Use the *port-id* option to specify the port for with you want to clear the counters.

Specify the **switched** keyword to clear counters for the switched packets. If the **switched** keyword is not specified the counters for both the routed packets and switched packets will be cleared. The **switched** keyword is accepted only if the routed packets and switched packets are counted separately.

Clearing extended counters statistics on a port

To clear all extended counters statistics simultaneously for a single port, enter the following command.

```
device# clear statistics ethernet 1/2 extended counters
```

Syntax: clear statistics ethernet *port_id orrange* **extended counters**

Use the *port_id or range* option to specify the port or range for which you want to clear the extended counters.

Clearing extended counters statistics on specific slot

To clear all extended counter statistics simultaneously for a slot, enter the following command.

```
device# clear statistics slot 2 extended counters
```

Syntax: clear statistics slot *slot-id* **extended counters**

Use the *slot-id* option to specify the slot number for which you want to clear the extended counters.

IP interface commands

You can display and clear the counter details of the physical and virtual IP interfaces.

Displaying IP interface counters

You can display aggregate count of the routed packets and switched packets of an IP interface using the following command.

```
<< If Routed/Switched separate mode >>
device# show ip interface ve 10 statistics
Extended Routed Counters (only applicable for G2 modules):
Total     RxPkts           TxPkts           RxBytes          TxBytes
          0                0                0                0
<< If Routed/Switched combined mode>>
device# show ip interface ve 10 statistics
Extended Routed/Switched Counters (only applicable for G2 modules):
Total     RxPkts           TxPkts           RxBytes          TxBytes
                     0                0                0                0
```

Syntax: show ip interface ethernet *port-id* **statistics**

Specify the *port id* of the interface for which you want to display the routed and switched packets aggregate count.

TABLE 31 show ip interface command output details

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying IP virtual interface counters

To display the counters for each physical port of a virtual IP interface, use the following command.

```
device#show ip interface ve 10 statistics ethernet 12/1
Extended Routed Counters (applicable for G2 modules only):
```

```
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0              0                0                0
     p0   0              0                0                0
     p1   0              0                0                0
                                <snip>
     p6   0              0                0                0
     p7   0              0                0                0
```

Syntax: show ip interface ve *vid* **statistics** [**ethernet** *port-id*]

Specify the *port id* of the virtual interface for which you want to display.

TABLE 32 show ip interface ve statistics command output details

Field	Description
Interface	The interface the counter statistics are displayed.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying detailed IP virtual interface counters

To display the detailed aggregate count of the routed packets and switched packets of a virtual IP interface are configured separate mode, use the following command.

```
device# show ip interface ve 10 statistics detail
Extended Routed Counters (applicable for G2 modules only):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0              0                0                0
     p0   0              0                0                0
     p1   0              0                0                0
                                <snip>
     p6   0              0                0                0
     p7   0              0                0                0
eth 12/2  0              0                0                0
     p0   0              0                0                0
                            <snip>
     p7   0              0                0                0
```

To display the detailed aggregate count of the routed packets and switched packets of a virtual IP interface are configured combined mode, use the following command.

```
device# show ip interface ve 10 statistics detail
Extended Routed/Switched  Counters (applicable for G2 modules only):
Interface RxPkts           TxPkts           RxBytes          TxBytes
eth 12/1  0              0                0                0
     p0   0              0                0                0
     p1   0              0                0                0
                                <snip>
     p6   0              0                0                0
     p7   0              0                0                0
```

Syntax: show ip interface ve *vid* **statistics** [**detail**]

Use the *vid* option to specify the interface name of the virtual IP interface for which you want to display the routed and switched packets aggregate count.

TABLE 33 show ip interface ve output details

Field	Description
Interface	The interface the counter statistics are displayed.

TABLE 33 show ip interface ve output details (continued)

Field	Description
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing IP interface counters

When clearing IP interface counters, the counters that are cleared are dependent on the accounting mode configuration. If you have configured accounting mode to count routed and switched packets separately, only the routed counters will be cleared. If you have configured accounting mode to count routed and switched packets together, the combined counter will be cleared.

To clear the routed packet and switched packet counters of a specific IP interface, enter the following command.

```
device# clear ip interface ethernet 1/2 statistics
```

Syntax: clear ip interface ethernet *port-id* **statistics**

Use the *port-id* option to specify the interface name to clear.

Clearing IP virtual interface counters

When clearing IP virtual interface counters, the counters that are cleared are dependent on the accounting mode configuration. If you have configured accounting mode to count routed and switched packets separately, only the routed counters will be cleared. If you have configured accounting mode to count routed and switched packets together, the combined counter will be cleared.

To clear the routed packet and switched packet counters of a specific virtual IP interface, enter the following command

device# clear ip interface ve 2 statistics

Syntax: clear ip interface ve *vid* **statistics**

Use the *vid* option to specify the virtual interface name to clear.

Transparent VLAN flooding

You can configure your Brocade device for transparent VLAN flooding. This feature allows packets to be forwarded without any form of CPU intervention including MAC learning and MAC destination lookups.

> NOTE
> Because this feature floods all VLAN packets in hardware, it is not expected to work in conjunction with routing functions such as establishing routing protocol neighbor and Layer 3 forwarding even when the VLAN has a VE configured.

> NOTE
> Enabling transparent VLAN flooding causes the bypass of the load balancing mechanisms of a Link Aggregation Group (LAG).

This implementation of transparent VLAN flooding has the following attributes:

- The ability to always distribute traffic to all members of a VLAN in hardware.
- It requires no CPU intervention and consequently can handle line-rate traffic forwarding.
- Because this feature does not use any MAC address entries in the CAM it is useful when MAC address entries need to be conserved.

- VLAN members can be tagged or untagged ports including a mix of tagged and untagged ports.
- The maximum number of Transparent VLAN Flooded instances is 4090.
- You can mix and match ports with different speeds.
- Other Layer 2 capabilities such as spanning tree are unaffected.
- Output Layer 3 ACLs may be associated with each port that is part of the VLAN instance being transparently flooded.
- You cannot configure a VE interface on a VLAN when Transparent VLAN Flooding is used

This feature is particularly useful in situations where MAC learning is not required for traffic forwarding. Examples of where this feature is useful include:

- A configuration where there are only 2 ports in a VLAN.
- Where traffic is looped back to a device through another VLAN for firewall or mirroring purposes.
- Where the number of MAC addresses will significantly overwhelm the memory and compute resources of a system.

NOTE
Packets that arrive on an interface with the same destination MAC address as the interface are forwarded in hardware just like packets with other destination addresses.

Enabling VLAN transparent forwarding

To enable VLAN transparent forwarding on VLAN 10, use the following command.

```
device(config)# vlan 10
device(config-vlan-10)# transparent-hw-flooding
```

Syntax: [no] transparent-hw-flooding

- Layer 2 inbound ACLs may be applied to any port in a VLAN that has TVF enabled.
- Layer 2 or Layer 3 outbound ACLs may be applied to any port in a VLAN that has TVF enabled.
- Input Layer 3 ACLs need to be applied to a virtual routing interface on the VLAN and should not be attached to a port directly. Note that the ACL would apply to all ports in the VLAN by default. If this is not desired, a subset of ports in the VLAN may be specified, as in the following configuration.

```
device(config)# interface ve 1
device(config-vif-1)# ip access-group 101 in ethernet 1/1
device(config-vif-1)# ip access-group ve-traffic
```

NOTE
The above example binds the ACL 101 to the virtual routing interface and configures the virtual routing interface to apply the input ACL 101 for switched and routed traffic received on port 1/1.

Enabling VLAN LAG load balancing

VLAN LAG load balancing allows for the transparent VLAN flooding feature to flood an outgoing LAG and load-balance correctly across all ports of the LAG.

TABLE 34 Hardware-enhanced VLAN module and FID pool size matrix

FID pool size —> Number of member ports	512	1024	2048	4096
2	256 (each VLAN requires 2 entries)	512 (Maximum of 480 load balancing instances are	1024 (Maximum of 480 load balancing instances are	2048

TABLE 34 Hardware-enhanced VLAN module and FID pool size matrix (continued)

FID pool size —> Number of member ports	512	1024	2048	4096
		supported by the hardware in 10Gx24 module.)	supported by the hardware in 10Gx24 module.)	The maximum number of TVF LAG load balancing instances is limited to 2016. (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)
4	128	256	512 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)	1024 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)
8	64	128	256	512 (Maximum of 480 load balancing instances are supported by the hardware in 10Gx24 module.)
16	32	64	128	256

TABLE 35 Software-enhanced limit for 48x1G and 24x1G modules

Ingress —> Egress	48x1G and 24x1G modules
48x1G and 24x1G modules	480 VLANs with up to 8 port member LAG

TABLE 36 Software-enhanced limit for 20x10G, 8x10G, 2x100G, and 4x40G modules

Ingress —> Egress	20x10G, 8x10G, 2x100G, and 4x40G modules
20x10G, 8x10G, 2x100G, and 4x40G modules	A maximum of 1024 VLANs with the 4-port member LAG A maximum of 256 VLANs with the 16-port member LAG

NOTE

Downgrading to prior releases may cause the pool size to be removed from the configuration and disabling of TVF VLAN LAG load balancing when the pool size is set to 4096.

To enable transparent VLAN LAG load balancing on VLAN 10, use the following commands.

```
device(config)# vlan 10
device(config-vlan-10)# transparent-hw-flooding lag-load-balancing
```

Syntax: [no] transparent-hw-flooding lag-load-balancing

NOTE

TVF VLAN LAG load balancing is applicable to Policy Based Routing (PBR) telemetry applications only.

Configuring TVF FID pool size

To configure maximum FID pool size for transparent VLAN flooding LAG load balancing globally, use this command.

device(config)# system-max tvf-lag-lb-fid-pool *number*

The *number* specifies the pool size values which are 0, 512, 1024, 2048, and 4096. Default is 0 (feature is disabled).
For information about the hardware-enhanced VLAN module/ FID pool size matrix and software-enhanced limit for different modules, refer to Enabling VLAN LAG load balancing on page 211.

Syntax: [no] system-max tvf-lag-lb-fid-pool *number*

NOTE
Downgrading the Brocade NetIron device to earlier releases may cause the pool size to be removed from the configuration when FID pool size is set to 4096.

Configuring TVF FID group size

To configure maximum FID group size for transparent VLAN flooding LAG load balancing globally, use this command.

device(config)#system-max tvf-lag-lb-fid-group *number*

The *number* specifies the group values which are 2, 4, 8, and 16. Default is 4.

This value determines the size of LAG allowed in the VLAN and LAG's with more ports are not allowed to use this value.

Syntax: [no] system-max tvf-lag-lb-fid-group *number*

Transparent VLAN flooding domain

The transparent VLAN flooding (TVF) domain provides an infrastructure to increase the overall egress traffic streams.

Currently Policy-based Routing (PBR) TVF uses flooding on VLANs for traffic streams and supports a maximum of 4090 egress traffic streams. With the TVF domain implementation, traffic can be flooded to the TVF domain by setting the TVF domain as a PBR next hop. The TVF domain supports 2016 TVF instances with LAG load balancing and, together with 4090 TVF instances without LAG load balancing, scales the overall egress traffic flow support to 6106.

NOTE
The TVF domain supports only TVF with LAG load balancing.

NOTE
A maximum of 480 load balancing instances are supported by the hardware in the 24x10G module. The egress streams with TVF LAG load balancing configured over 480 instances will be forwarded without LAG load balancing.

Configuring the TVF domain

The following steps configure the TVF domain and add member ports to the TVF domain.

1. Enter the **configure terminal** command to enter global configuration mode.

```
device# configure terminal
```

2. Enter the **tvf-domain** command to configure a TVF domain with a named TVF domain ID.

```
device(config)# tvf-domain 1 name domainuser
```

   Valid values for the TVF domain ID are from 1 through 2016. The name can be up to 64 characters in length.

3. Enter the **port** command to add member ports to the TVF domain.

```
device(config-tvf-domain-1)# port ethernet 1/1 ethernet 2/1
```

The number of ports in the LAG that can be added to the TVF domain is limited based on the maximum FID pool size configured using the **system-max tvf-lag-lb-fid-group** command.

Setting the TVF domain as a PBR next hop

The following steps configure the TVF domain as the next hop for a route map to support transparent VLAN flooding (TVF) with LAG load balancing.

Configure the required IPv4 ACLs and IPv6 ACLs to be added to the route map. For more information about configuring ACLs, refer to the *Brocade NetIron Security Configuration Guide*.

1. Enter the **configure terminal** command to enter global configuration mode.

```
device# configure terminal
```

2. Enter the **route-map** command to define the route and specify the match criteria and the resulting action if all of the match clauses are met.

```
device(config)# route-map test-route permit 99
```

3. Add IPv4 ACLs or IPv6 ACLs or both to match the IP address that is permitted by the ACL.

```
device(config-routemap test-route)# match ip address SGW_1_ACL
device(config-routemap test-route)# match ipv6 address SGW_2_ACL
```

4. Enter the **set next-hop-tvf-domain** command to configure a TVF domain as the next hop for a route map.

```
device(config-routemap test-route)# set next-hop-tvf-domain 1
```

Configuration example of TVF domain as PBR next hop for TVF with LAG load balancing

Complete the following steps to configure TVF domain as PBR next hop for TVF with LAG load balancing.

1. Configure system maximum requirements to support LAGs and TVF LAG load balancing.

```
device(config)# system-max trunk-num 32
device(config)# system-max tvf-lag-lb-fid-pool 4096
device(config)# system-max tvf-lag-lb-fid-group 4
```

2. Configure the TVF domain and assign the ports.

```
device(config)# tvf-domain 1
device(config-tvf-domain-1)# port ethernet 1/1 ethernet 2/1 ethernet 5/1 ethernet 6/1
```

3. Configure IPv6 ACLs and IPv4 ACLs to be routed using PBR.

```
device(config)# ipv6 access-list v6_Permit_Any
device(config-ipv6-access-list v6_Permit_Any)# permit ipv6 any any
device(config-ipv6-access-list-v6_Permit_Any)# exit
device(config)# ipv6 access-list v6_brc_Test_p000_bi_moof
device(config-ipv6-access-list v6_brc_Test_p000_bi_moof)# permit vlan 1001 ipv6 any any
device(config-ipv6-access-list v6_brc_Test_p000_bi_moof)# exit
device(config)# ip access-list extended v4_Permit_Any
device(config-ext-nacl-v4_Permit_Any)# permit ip any any
device(config-ext-nacl-v4_Permit_Any)# exit
device(config)# ip access-list extended v4_brc_Test_p000_bi_moof
device(config-ext-v4_brc_Test_p000_bi_moof)# permit vlan 1001 ip any any
```

4. Configure route maps and add the configured ACLs.

```
device(config)# route-map Mall permit 1001
device(config-routemap Mall)# rule-name brc_Test_p000_bi_moof
device(config-routemap Mall)# match ip address v4_brc_Test_p000_bi_moof
device(config-routemap Mall)# match ipv6 address v6_brc_Test_p000_bi_moof
```

5. Add the configured TVF domain as the next hop to the route map.

```
device(config-routemap Mall)# set next-hop-tvf-domain 1
```

6. Configure a VLAN and add the interfaces required for the TVF LAG and LAG load balancing.

```
device(config)# vlan 1 name DEFAULT-VLAN
device(config-vlan-1)# no untagged ethernet 1/1 ethernet 2/1 ethernet 5/1 ethernet 6/1
device(config-vlan-1)# no spanning-tree
```

7. Create a LAG and add the interfaces.

```
device(config)# lag blue static
device(config-lag-blue)# ports ethernet 5/1 ethernet 6/1
device(config-lag-blue)# primary-port 5/1
device(config-lag-blue)# deploy
```

8. Apply PBR to the ingress interfaces.

```
device(config)# interface ethernet 11/1
device(config-if-e10000-11/1)# qos multicast shaper best-effort rate 10000
device(config-if-e10000-11/1)# qos multicast shaper guaranteed rate 100000000
device(config-if-e1000-11/1)# enable
device(config-if-e1000-11/1)# ip policy route-map Mall
device(config-if-e1000-11/1)# ipv6 policy route-map Mall
device(config-if-e1000-11/1)# allow-all-vlan pbr
```

Displaying TVF domain information

Use the **show tvf-domain** command to view details of the TVF domain configuration, such as the TVF domain ID, ports added to the TVF domain, system maximum requirements to support TVF LAG load balancing, and so on.

The following example displays information about the TVF domain.

```
device(config)# show tvf-domain
**** TVF LAG Load Balancing ****
TVF LAG Load Balancing is enabled!
TVF FID pool size: 4096, Max FID groups: 2048, FID group size: 2
2047 (VLAN 32, TVF Domain 2015) TVF LAG Load balancing groups are configured
TVF LAG Load balancing FID programming is done

TVF domain memory usage : 735848 bytes
Per entry usage         : 365 bytes

TVF Domain ID 1, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 2, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 3, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 4, Name [None]
Ports : ethe 8/5 to 8/8

TVF Domain ID 6, Name [None]
Ports : ethe 8/5 to 8/8
```

The following example displays the information of a specific TVF domain.

```
device(config)# show tvf-domain 1
TVF Domain ID 1, Name [None]
Ports : ethe 8/5 to 8/8
------------------------------------------------------------
Port  Type      Protocol  State
8/5   TRUNK     NONE      UP
8/6   TRUNK     NONE      UP
8/7   TRUNK     NONE      UP
8/8   TRUNK     NONE      UP
Group ID: 33, FID Base 0x00009ffe, FID Count 2
tvf_lag_lb_fid0: 0x00009ffe, mask ethe 8/5 ethe 8/7
tvf_lag_lb_fid1: 0x00009fff, mask ethe 8/6 ethe 8/8
```

The following example displays a brief summary of all the configured TVF domains.

```
device(config)# show tvf-domain brief
**** TVF LAG Load Balancing ****
TVF LAG Load Balancing is enabled!
TVF FID pool size: 4096, Max FID groups: 2048, FID group size: 2
2047 (VLAN 32, TVF Domain 2015) TVF LAG Load balancing groups are configured
TVF LAG Load balancing FID programming is done

TVF domain memory usage : 735848 bytes
Per entry usage         : 365 bytes
TVF   Name       Ports
----- ----       -----
1     [None]     Ports : ethe 8/5 to 8/8
2     [None]     Ports : ethe 8/5 to 8/8
3     [None]     Ports : ethe 8/5 to 8/8
4     [None]     Ports : ethe 8/5 to 8/8
6     [None]     Ports : ethe 8/5 to 8/8
7     [None]     Ports : ethe 8/5 to 8/8
8     [None]     Ports : ethe 8/5 to 8/8
9     [None]     Ports : ethe 8/5 to 8/8
10    [None]     Ports : ethe 8/5 to 8/8
11    [None]     Ports : ethe 8/5 to 8/8
12    [None]     Ports : ethe 8/5 to 8/8
13    [None]     Ports : ethe 8/5 to 8/8
14    [None]     Ports : ethe 8/5 to 8/8
15    [None]     Ports : ethe 8/5 to 8/8
16    [None]     Ports : ethe 8/5 to 8/8
17    [None]     Ports : ethe 8/5 to 8/8
18    [None]     Ports : ethe 8/5 to 8/8
19    [None]     Ports : ethe 8/5 to 8/8
20    [None]     Ports : ethe 8/5 to 8/8
21    [None]     Ports : ethe 8/5 to 8/8
22    [None]     Ports : ethe 8/5 to 8/8
23    [None]     Ports : ethe 8/5 to 8/8
24    [None]     Ports : ethe 8/5 to 8/8
25    [None]     Ports : ethe 8/5 to 8/8
26    [None]     Ports : ethe 8/5 to 8/8
27    [None]     Ports : ethe 8/5 to 8/8
28    [None]     Ports : ethe 8/5 to 8/8
29    [None]     Ports : ethe 8/5 to 8/8
30    [None]     Ports : ethe 8/5 to 8/8
31    [None]     Ports : ethe 8/5 to 8/8
32    [None]     Ports : ethe 8/5 to 8/8
33    [None]     Ports : ethe 8/5 to 8/8
34    [None]     Ports : ethe 8/5 to 8/8
35    [None]     Ports : ethe 8/5 to 8/8
36    [None]     Ports : ethe 8/5 to 8/8
37    [None]     Ports : ethe 8/5 to 8/8
```

The following example displays detailed information of each TVF domain.

```
device(config)# show tvf-domain detail
**** TVF LAG Load Balancing ****
TVF LAG Load Balancing is enabled!
TVF FID pool size: 4096, Max FID groups: 2048, FID group size: 2
2047 (VLAN 32, TVF Domain 2015) TVF LAG Load balancing groups are configured
TVF LAG Load balancing FID programming is done

TVF domain memory usage : 735848 bytes
Per entry usage         : 365 bytes

TVF Domain ID 1, Name [None]
Ports : ethe 8/5 to 8/8
----------------------------------------------------------
Port  Type      Protocol  State
8/5   TRUNK     NONE      UP
8/6   TRUNK     NONE      UP
8/7   TRUNK     NONE      UP
8/8   TRUNK     NONE      UP
Group ID: 34, FID Base 0x00009ffe, FID Count 2

TVF Domain ID 2, Name [None]
Ports : ethe 8/9 to 8/12
----------------------------------------------------------
Port  Type      Protocol  State
8/9   TRUNK     NONE      UP
8/10   TRUNK     NONE      UP
8/11   TRUNK     NONE      UP
8/12   TRUNK     NONE      UP
Group ID: 33, FID Base 0x00009ffe, FID Count 2
```

The following example displays the details of the port configured in the TVF domain.

```
device(config)# show tvf-domain ethernet 8/6
TVF Domain : 1
TVF Domain : 2
TVF Domain : 3
TVF Domain : 4
TVF Domain : 6
TVF Domain : 7
TVF Domain : 8
TVF Domain : 9
TVF Domain : 10
TVF Domain : 11
TVF Domain : 12
TVF Domain : 13
TVF Domain : 14
TVF Domain : 15
TVF Domain : 16
TVF Domain : 17
TVF Domain : 18
TVF Domain : 19
TVF Domain : 20
TVF Domain : 21
TVF Domain : 22
TVF Domain : 23
TVF Domain : 24
```

Transparent firewall mode

The transparent firewall mode allows the device to switch control packets destined to itself. By default, Brocade devices will drop control packets received with the device's MAC address as the packet's destination MAC address (that is, packets destined to the switch or router). Under the transparent firewall mode, switching packets destined to itself is allowed. The transparent firewall mode feature is a per VLAN configuration and is disabled by default.

Enabling a transparent firewall

To set the mode to transparent, enter a command such as the following.

```
device(config-vlan-10)# transparent-fw-mode
```

To set the mode to routed, enter a command such as the following.

```
device config-vlan-10)# no transparent-fw-mode
```

Syntax: [no] transparent-fw-mode

NOTE
Transparent firewall mode is available only on the Brocade NetIron CES and Brocade NetIron CER devices.

Displaying VLAN information

After you configure the VLANs, you can view and verify the configuration using the commands discussed in this section.

Displaying VLAN information

Use the **show vlan** command under the vlan-policy configuration.

NOTE
VLAN byte counters are displayed in the output of the **show vlan** command on an MPLS enabled VE interface.

```
device (config)# show vlan
Configured PORT-VLAN entries: 4
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level -,
Priority Force 0
Topo HW idx : 0 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : NONE
Untagged Ports : ethe 1/1 to 1/48 ethe 2/1 to 2/2
Associated Virtual Interface Id: NONE
PORT-VLAN 100, Name [None], Priority Level -, Priority
Force 0
Topo HW idx : 2 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : ERP
Tagged Ports : ethe 1/1 ethe 1/10 ethe 1/29
Associated Virtual Interface Id: NONE
PORT-VLAN 200, Name [None], Priority Level -, Priority
Force 0
Topo HW idx : 0 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : NONE
Tagged Ports : ethe 1/1 ethe 1/10 ethe 1/29
Associated Virtual Interface Id: 120
PORT-VLAN 4095, Name CONTROL-VLAN, Priority Level -,
Priority Force 0
Topo HW idx : 1 Topo SW idx: 257 Topo next
vlan: 0
L2 protocols : NONE
Associated Virtual Interface Id: NONE
```

Syntax: show vlan [vlan-id] [brief | detail] [Ethernet slot/port] [begin expression | exclude expression | include expression]

The output shows the following information.

TABLE 37 Output of show vlan

This field...	Displays...
Configured PORT-VLAN entries	Number of port-based VLANs in the configuration.
Maximum PORT-VLAN entries: 4090	Maximum number of port-based VLANs that you can configure.
Default PORT-VLAN id	ID of the default VLAN.
PORT-VLAN	ID of the port-based VLAN
Name	Name of the port-based VLAN. [None] appears if a name has not been assigned.
Priority Level	Priority level assigned to the port-based VLAN
Priority Force	The priority force is configured on the ingress port with a priority value between 0 and 7. The default is 0. The priority force option allows you to force the priority on the ingress pram, or choose not to enforce the priority.
Topo HW idx	A topology hardware index is a unique hardware ID that is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The range for hw-index is 0 - 511.
Topo SW idx	The topology group id associated with the VLAN.
Topo next vlan	Next VLAN id in the topology group.
L2 protocols	Layer 2 control protocol configured on the VLAN
Untagged or Tagged Ports	ID of the untagged or tagged ports that are members of the VLAN
Associated Virtual Interface Id	The ve interface id is displayed when a router-interface is configured for the VLAN. If no router-interface is configured, the field displays NONE.
Bytes received	Displays the number of received bytes across all ports for a specified VLAN. By default, the vlan-accounting command is turned on and hence the Bytes received field is also displayed by default. However, if VLAN accounting for a VLAN is turned off, then the Bytes received field is not displayed in the output. For more information on enabling VLAN byte counters, refer to Extended VLAN counters for 8x10G modules on page 203.

To display information for a specific VLAN, enter a VLAN id as shown in the example below.

```
device(config-vlan-13)#show vlan 2001
PORT-VLAN 2001, Name [None], Priority Level 0, Priority Force 0
Topo HW idx    : 0    Topo SW idx: 257    Topo next vlan: 0
L2 protocols   : MRP
Tagged Ports   : ethe 2/1 to 2/6 ethe 2/11 to 2/14 ethe 2/23 to 2/24
Untagged Ports : ethe 1/1 ethe 1/5
Associated Virtual Interface Id: NONE
```

Displaying VLAN information for specific ports

To determine which VLANs a port is a member of, enter the following command.

```
device# show vlan e 4/1
VLANS 1
VLANs 100
```

Syntax: show vlan ethernet slot-number/port-number [| [**begin expression** | **exclude expression** | **include expression**]

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

The output shows the following information.

TABLE 38 Output of show vlan Ethernet

This field...	Displays...
VLANs	The IDs of the VLANs that the port is a member of.

Displaying VLAN status and port types

To display detailed information about the state, port types, port modes, of a VLAN, as well as control protocols configured on the VLAN, enter the following command.

```
device# show vlan detail
Untagged Ports  : ethernet 2/1 to 2/20 ethernet 4/4
Tagged Ports    : None
Dual-mode Ports : ethernet 3/1 to 3/20 ethernet 4/1 to 4/3
Default VLAN    : 1
Control VLAN    : 4095
VLAN Tag-type   : 0x8100
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
2/1   PHYSICAL  UNTAGGED  NONE      DISABLED
2/2   PHYSICAL  UNTAGGED  NONE      DISABLED
2/3   PHYSICAL  UNTAGGED  NONE      DISABLED
2/4   PHYSICAL  UNTAGGED  NONE      DISABLED
2/5   PHYSICAL  UNTAGGED  NONE      DISABLED
.
. (output edited for brevity)
.
4/1   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/2   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/3   PHYSICAL  UNTAGGED  NONE      FORWARDING
4/4   PHYSICAL  UNTAGGED  NONE      DISABLED
PORT-VLAN 100, Name [None], Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
4/1   PHYSICAL  TAGGED    STP       FORWARDING
4/2   PHYSICAL  TAGGED    STP       BLOCKING
```

Syntax: show vlan [**detail** | **brief**] [**begin expression** | **exclude expression** | **include expression**]

The output shows the following information.

TABLE 39 Output of show vlan detail

This field...	Displays...
Untagged Ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as untagged ports in all the VLANs on the device.
Tagged Ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as tagged ports in all the VLANs on the device.
Dual-mode ports	This line appears if you do not specify a VLAN. It lists all the ports that are configured as dual-mode ports in all the VLANs on the device.
Default VLAN	ID of the default VLAN
Control VLAN	ID of the control VLAN
PORT-VLAN #, Name, Priority Level	Information for each VLAN in the output begins with the VLAN type and its ID, name and priority level. Then ports that are members of the VLAN are listed, with the following information:
Port	Port slot-number/port-number
Type	Port type: physical or LAG
Tag-Mode	Tag mode of the port: untagged, tagged, or dual-mode
Protocol	Protocol configured on the VLAN.

TABLE 39 Output of show vlan detail (continued)

This field...	Displays...
State	Current state of the port such as disabled, blocking, forwarding, etc.

Displaying VLAN group information

To display information about VLAN groups, enter the following command.

```
device# show vlan-group 10
Configured VLAN-Group entries : 1
Maximum VLAN-Group entries : 32
VLAN-GROUP 10
Number of VLANs: 4
VLANs: 10 to 13
Tagged ports: ethernet 3/1
```

Syntax: show vlan-group [**vlan-group-id**] [**begin** *expression* | **exclude** *expression* | **include** *expression*]

The output shows the following information.

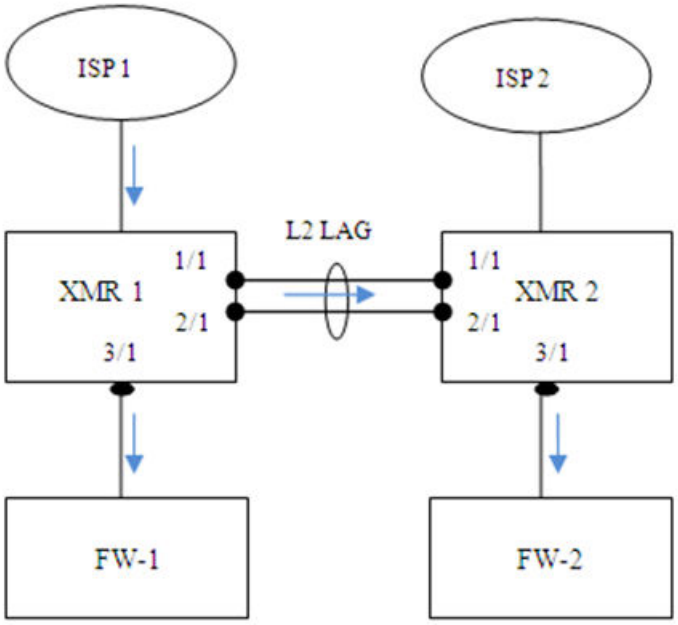
TABLE 40 Output of show vlan Ethernet

This field...	Displays...
Configured VLAN-Group entries	Number of VLAN groups that have been configured on the device.
Maximum VLAN-Group entries	Maximum number of VLAN groups that can be configured on the device.
VLAN-Group #	ID of the VLAN group
VLANs	VLANs that belong to the VLAN group.
Tagged ports:	Type and ID of the tagged ports that are members of the VLAN group

Multi-port static MAC address

The multi-port static MAC address feature enables you to send traffic destined for a particular MAC address on a set of ports of a VLAN instead of flooding the traffic on all ports of the VLAN. This feature enables you to configure a Layer 2 static multicast MAC address. This feature is supported on both the Brocade NetIron XMR Series, Brocade NetIron MLX Series and Brocade NetIron CER Series, Brocade NetIron CES Series series platforms.

FIGURE 18 Multi-port static MAC address example



Consider two Brocade NetIron XMR Series switches, XMR 1 and XMR 2, connected by a Layer 2 Link Aggregation Group (LAG), with member ports 1/1 and 2/1, as shown in Figure 18. To send traffic to a multicast MAC 'M' to the 3/1 ports on XMR 1 and XMR 2, you can create a static multicast MAC 'M' on the 3/1 ports and Layer 2 LAG (primary port 1/1 of Layer 2 LAG). Traffic sent to multicast MAC 'M' from ISP1 will be sent to ports 1/1 and 3/1 on XMR 1; and XMR 2 will send the traffic received on port 1/1 to FW-2, connected to port 3/1, instead of flooding the traffic on all ports of the VLAN.

Configuring multi-port static MAC address

To configure multi-port static MAC address, enter the following command at the VLAN configuration node level of the CLI. You must configure at least one port.

```
device# vlan 10
static-mac-address 0100.5e42.7f40 multi-ports ethe 1/1 to 1/3
ethe 2/1 to 2/5 priority 5
```

Syntax: [no] static-mac-address *mac-address* **multi-ports ethernet** [*slot1/port1* | [*slot1/port1* **to** *slot1/port#k*] .. **ethernet** [*slot#n/port#n* **to** *slot#n/port#m*] [**priority** *0-7*]

Limitations

The configuration of multi-port static MAC address has the following limitations:

- A maximum number of 400 multi-port MAC addresses and static MAC addresses can be configured in a system; however, the number of configured entries is limited by the number of multicast FIDs available in the system.
- FIDs with the same port mask are shared among multiple multi-port MAC addresses and multi-port ARP entries to conserve the number of multicast FIDs created in the system.
- Multi-port static MAC address cannot be configured on VLAN groups.

- You cannot add any of the interface MAC address as multi-port static MAC address.
- Multi-port static MAC address can be configured for either unicast or multicast addresses.
- Unicast MAC addresses configured as multi-port static MAC addresses will not be learned dynamically in the system or allowed to be dynamically moved to a different port.
- If the multi-port static MAC address being added already exists in the dynamic MAC table, then the dynamic MAC will be deleted and replaced with the configured multi-port static MAC.
- Trunk load-balancing is not supported. The multi-port static MAC address traffic will always be forwarded on the active primary port of the trunk port.
- The multi-port MAC feature is supported in a pure Layer 2 forwarding vlan to forward Layer 2 traffic to multiple ports. This feature should not be configured on a vlan with a virtual router-interface.

Error messages

Error messages are displayed in the following cases:

- You can configure multi-port static MAC addresses only if all the ports in the port list are members of the VLAN.

```
device(config-vlan-100)#static-mac 0001.0001.0003 multi-port eth 2/11
Error - Multiport mac cannot be configured, Port 2/11 is not member of vlan 100
```

- You must provide the complete port mask for deleting the multi-port static MAC address configuration.

```
device(config-vlan-100)#no static-mac 0001.0001.0002 multi-ports eth 2/19 to 2/20
Error - the port list does not match with the configured multiport mac port list
```

- On a trunk port, multi-port static MAC address configuration is allowed only on the primary port of the trunk.

```
device(config-vlan-100)#static-mac-address 0001.0001.0003 multi-ports ethe 2/19 to 2/20 ethe 4/3
Error - Multiport mac cannot be configured with non-primary trunk port  4/3
```

- A port with multi-port static MAC address configuration cannot be a secondary member of the trunk group.

```
device(config-lag-LAG1)#ports eth 4/11
Error: port 4/11 is part of multiport-mac and cannot be added as secondary port of a trunk
```

- When a LAG primary port is part of a multi-port static MAC address, the LAG cannot be undeployed or deleted. Also, when a non-LAG port is part of a multi-port static MAC address, you cannot deploy a LAG with that port. These configurations will be rejected. The following are the sample error messages for each of these cases:
  - Veto check for deploying LAG.

```
device(config-lag-LAG1)#deploy
Error: LAG LAG1 primary port 4/11 is configured as part of a multi-port mac entry, cannot deploy the LAG
```

- – Veto check for undeploying LAG.

```
device(config-lag-LAG2)#no deploy
Error: The primary port 4/15 is configured as part of a multi-port mac entry, cannot undeploy the LAG
```

- – Veto check for deleting LAG.

```
device(config)#no lag LAG2
Error: The primary port 4/15 is configured as part of a multi-port mac address, cannot remove the LAG
```

- A port belonging to a multi-port static MAC address is not allowed to be removed from a VLAN unless it is removed from all the multi-port static MAC addresses.

```
device(config-vlan-100)#no tag e 2/19
Error - port 2/19 is configured as part of a multi-port mac address, cannot remove port from vlan
```

- Module configuration deletion (no module) will be rejected if any of the ports in that module are configured as part of any multi-port static MAC addresses.

```
device(config)#no mod 4 ni-mlx-20-port-1g-100fx
Error - module 4 has ports that are member of the multi-port mac addresses, Cannot remove the module
```

Displaying multi-port static MAC address information

You can display the following information about multi-port static MAC addresses on the device:

- Running configuration
- Changes in the MAC table
- M-port debug information
- LP information

Displaying running configuration

To display the running configuration information of multi-port static MAC addresses, enter the following command.

```
device# show run
vlan 10
tagged ethe 1/1 to 1/20 ethe 2/1 to 2/48
static-mac-address 0100.5e42.7f40 multi-ports ethe 1/1 to 1/3
ethe 2/1 to 2/5 prority 5
```

Displaying changes in the MAC table

You can display the complete MAC table, a specific entry in the MAC table, or a specific MAC entry with M-port details.

To display the complete MAC table, enter the following command.

```
device# show mac
MAC Address         Port          Age          VLAN    Type
0100.5e42.7f40          1/1...        Static          10
Ports  : e 1/1 to 1/3 e 2/1 to 2/5
0000.999d.9996      2/20            Static        10
```

To display a specific MAC address from the MAC table, enter the following command.

```
device# show mac 0100.5e42.7f40
MAC Address         Port          Age          VLAN    Type
0100.5e42.7f40          1/1...       Static       10
Ports: e 1/1 to 1/3 e 2/1 to 2/5
```

To display a specific MAC address with M-port details, enter the following command.

```
device#show mac 0100.5e42.7f40 debug
MAC Address         Port          Age          VLAN    Type
0100.5e42.7f40          1/1...          Static       10
Mport: 30324 FID: 0x00008006 Ports: e 1/1 to 1/3 e 2/1 to 2/5
```

SA and DA learning and aging

Static MAC addresses and multi-port MAC addresses can always be programmed in the CAM of all LP modules with valid VLAN membership. Multi-port static MAC addresses are added or removed from the CAM when a port is added to or deleted from a VLAN. These addresses cannot be dynamically learned or moved to a different port. These addresses will not be removed from the hardware unless the user deletes the multi-port static MAC addresses.

MP switchover and hitless upgrade

The Multi-port static MAC Address feature supports MP switchover and hitless upgrade.

The static MAC table is maintained on both active and standby MPs. The static MAC table is synched to standby MP by CLI configuration commands. The M-port table and FID are also synched from the active MP to the standby MP. After MP switchover, M-port MAC addresses are associated with the FID and, in the case of a missing FID, a new FID will be created and programmed for the multi-port static MAC address.

The static MAC table and FID table will be reprogrammed after LP reload.

Flooding features

User-configured multi-port static MAC addresses will always be programmed on DA CAMs in all PPCR CAMs. So, traffic with this DA MAC address will never be flooded in the VLAN, even when flooding features like transparent VLAN, unknown unicast flooding, multicast flooding, and CPU protection are configured on the system.

ESI overview

An Ethernet Service Instance (ESI) is a provisioning environment for defining VLAN and other layer 2 parameters for creating services, typically across a carrier network.

In a local area network a total of 4K VLANs can be configured across the entire network domain. With a Q-in-Q bridging, VLANs from the set of 4K VLANs can be inter-connected across a provider network. While ESI allows a carrier to provide transport services for different sets of 4K VLANs for different customers, the provider network is still limited to using 4K VLANs across all of the customers connected to a single box, as it is very difficult to configure and manage different sets of 4K VLANs across the different ports within a single system.

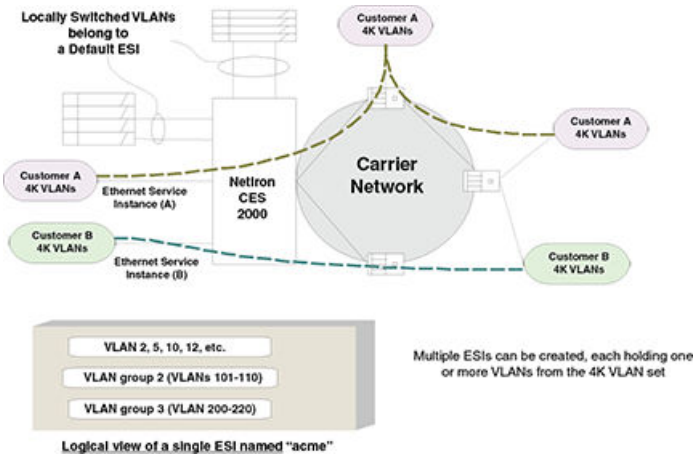
Using an ESI, a carrier can create service instances that hold one or more VLANs. Each instance has an alphanumeric name that is locally unique. The purpose of creating an instance is to provide a container to hold VLANs and other layer2 parameters that define properties of all of the elements contained within the instance.

VLANs are added to an ESI using a standard VLAN command for individual VLANs or a VLAN group command for adding a set of VLANs.

In a simplified network shown in ESI overview, customers A and B are connected to a Brocade NetIron CES Series device with each customer having a separate set of 4K VLANs. One or more ESIs are created to hold these 4K VLANs.

> NOTE
> Although theoretically it is possible to add sets of 4K VLANs in ESIs, the actual number of VLANs in an ESI is limited by the use of memory and hardware resources.

Ethernet Service Instance for VLAN configuration



Once an ESI is defined, Brocade NetIron CES Series and Brocade NetIron CER Series devices operate on rules for configuring VLANs inside an ESI, and check against configuration incompatibilities (such as configuring the same VLAN value from two different ESIs on the same port).

Types of ESI

There are two types of Ethernet Service Instances, as described:

Default ESI

In ESI overview on page 225, VLANs associated with ports in the top left corner of the Brocade NetIron CES Series and Brocade NetIron CER Series devices aren't being transported over to the carrier network - these VLANs are being locally switched and connected with switches in the local area network. The Brocade NetIron CES Series and Brocade NetIron CER Series devices support 4K VLANs of this type, without any ESI configured. Internally, these VLANs are associated with a Default ESI, and are referred to as 'Regular VLANs'.

Customer ESI

ESIs that are configured to hold customer VLANs that need to be transported across a carrier network are usually referred to as customer ESIs. A customer ESI always has a Layer 2 protocol VLAN encapsulation applied to all VLANs in the ESI.

In a carrier network, an incoming customer VLAN packet will usually be configured with successive encapsulations, such as with service VLANs, in-service identifiers, or backbone VLANs. Each encapsulation is associated with a different ESI. To define the encapsulation hierarchy, an ESI for an incoming packet is defined as a client of the ESI for the next encapsulation. The next encapsulation ESI is referred to as a 'provider ESI' and the ESIs that are declared as client ESIs are referred to as 'client ESIs'.

Depending on their association, customer ESIs can be one of the three types:

- Standalone ESI - An ESI that is not linked to any other ESI, and are used only to hold VLANs and define their properties.
- Provider ESI - An ESI with one VLAN, and one or more client ESIs, each holding one or more VLANs.
- Client ESI - An ESI that is defined to be a client of another ESI, and that can have one or more VLANs defined inside it.

Configuration considerations

The following rules apply for CLI operations for Provider Bridge (PB) and related protocols:

- To prevent topology changes at startup, it is recommended that you not use the same ESI-Vlan ID as the Default-Vlan ID.
- An ESI is created for each service (such as a customer CVLANs, SVLANs, PBB, etc.).
- All attributes for an ESI - such as VLAN, port binding, encapsulation, etc., are defined inside an ESI.
- To prevent configuration errors, no parameter overrides are permitted outside of an ESI.
- ESIs can be nested to provide multiple protocol encapsulations for a packet. Restrictions on bindings can be present depending on the actual platform. When nesting is used, inner ESIs are called client ESIs.
- A given VLAN means CVLAN or SVLAN, depending on the encapsulation definition for the ESI:
  - If no encapsulation is defined as "cvlan" the VLAN refers to CVLAN.

    NOTE
    For ESI VLANs, It is mandatory to define an encapsulation.

-
  - If encapsulation is "svlan", the VLAN refers to SVLAN (PB).
  - If encapsulation is "bvlan", the VLAN refers to B-VLAN (PBB).
- ISID values are treated differently from other VLANs, since ISID parameters have no networking association and are used only for mapping different SVLANs into service identifiers.

Creating an ESI

Create an ESI by naming it and specifying encapsulation type for all the VLANs inside it.

Creating an ESI with CVLAN tagging

To create an ESI with CVLAN tagging named "acme", enter a command such as the following.

```
device (config)# esi acme encapsulation cvlan
```

Syntax: [no] esi *esi-name* **encapsulation cvlan** | **svlan** | **isid** | **bvlan**

Use the **cvlan** parameter to specify the encapsulated customer VLAN (CVLAN).

Use the **svlan** parameter to specify the encapsulated service VLAN (SVLAN).

Use the **isid** parameter to specify the encapsulation for the mapping of SVLANs into service identifiers.

Use the **bvlan** parameter to specify the encapsulated backbone VLAN (BVLAN).

Once an ESI is created, subsequent invocations of the ESI do not require the encapsulation parameter.

Defining CVLANs inside the ESI

To define CVLANs inside the ESI, enter a command such as the following.

```
device(config-esi-acme)# vlan 10
```

Configuring the CVLAN to be tagged

To configure CVLAN 10 to be tagged on port 1/1, enter a command such as the following.

```
device(config-esi-acme-vlan-10)# tagged ethernet 1/1
```

Show VLAN commands

The following **show** commands will display custom ESI configurations for VLANs.

Displaying information for a VLAN inside an ESI

To display a VLAN inside an ESI, enter a command such as the following.

```
device(config)#show esi acme vlan 10
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
VLAN 10 details:
----------------
PORT-VLAN 10, Name [None], Priority Level-,Priority Force 0
L2 protocols   : NONE
ESI: bay Encapsulation: cvlan
----------------
No ports associated with VLAN
Arp Inspection: 0
DHCP Snooping: 0
L2 protocol forwarding mode:Tunnel
Flood domain ID 4176
```

Syntax: show vlan *num*

Displaying information for a VLAN inside an ESI in brief format

The **show vlan brief** command displays VLANs in a tabular format for compactness. This command may be executed from any CLI level.

```
device#show vlan brief
Configured PORT-VLAN entries: 1
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1
VLAN     Name       Encap ESI                              Pri Ports
----     ----       ----- ---                              --- -----
10       [None]     cvlan acme                             -
```

Syntax: show vlan brief

Displaying a single ESI

To display details of a single ESI, enter the following command from any level of the CLI.

```
device#show esi acme
ESI Name        Encap    Number of  Provider        Provider   Provider Client
                         Members    ESI             Encap      VLAN     ESIs
--------        -----    ---------  --------        -----      -------- ------
acme            cvlan    1                                           0
VLAN(s) at this ESI:
--------------------
VLAN     Name       Pri [L2 Protocols]       Ports
10       [None]     cvlan acme                             -   NONE
```

Displaying all ESIs

Use the **show esi** command to display a list of all the ESIs configured in the system. This command can be used at any level of the CLI.

```
device(config)#show esi
ESI Name        Encap    Number of  Provider        Provider   Provider Client
                         Members    ESI             Encap      VLAN     ESIs
--------        -----    ---------  --------        -----      -------- ------
acme            cvlan    1                                           0
```

Syntax: show esi *name*

Tag-type configuration

For the Brocade NetIron CES Series and Brocade NetIron CER Series, the following two VLAN tag-types are allowed that can be configured globally:

- **tag1** applies to customer edge ports (CVLAN) by default.
- **tag2** applies to provider-network, backbone-edge, and backbone-network port types (SVLAN and BVLAN) by default.

NOTE
The **tag1** and **tag2** are independent of port-types, so the system can be configured to use **tag1** for SVLAN, BVLAN and **tag2** for CVLAN.

Configuring tag-types

You can set the ISID value using a separate command similar to the Brocade NetIron XMR Series and Brocade NetIron MLX Series command as shown below.

Syntax: [no] tag-value isid *num*

You can configure CVLAN, SVLAN, and BVLAN tag-types as shown below.

```
device(config)# tag-value tag1 8100
device(config)# tag-value tag2 9100
device(config)# tag-type cvlan tag1 svlan tag2 bvlan tag2
```

Syntax: [no] tag-value *num*

Syntax: tag-type *tag-n*

The *num* parameter specifies the value assigned to the tag. The default value for **tag1** is 0x8100 and for **tag2** is 0x88a8.

The *tag-n* parameter can be either **tag1** or **tag2** .

Tag type can be changed from a default value to a specific port as shown in the following example.

```
device(config-if-e1000-1/1)# tag-type tag2 ethernet 1/1
device(config-if-e1000-1/1)# tag-type tag1 ethernet 1/2
```

Syntax: tag-type *tagid* **ethernet** *interface_id*

The *tagid* parameter can be either **tag1** or **tag2** . Possible tagid values are:

- – isid - to set the isid tag-value
  – tag1 - to set the tag-type tag1 value
  – tag2- to set the tag-type tag2 value

The *interface_id* parameter specifies the Ethernet slot and port ID.

Restrictions

The tag-type has the following restrictions:

- CVLAN and SVLAN cannot have the same tag-type but the tag-value can be set to the same.
- SVLAN and BVLAN must have the same tag-type.
- Port-type must be set to the default to configure the port-level tag-type.

Displaying tag types

To display the different tag types, enter a command such as the following.

```
device(config)#show tag-type
Encap      Current VLAN Tags             Default VLAN Tags
-----      -----------------             -----------------
cvlan                 8100                       8100
svlan                 9100                       88A8
isid                  86B5                       88E7
bvlan                 9100                       88A8
```

NOTE

The Brocade NetIron CES Series and Brocade NetIron CER Series require that the SVLAN and BVLAN tag types be same. The default tag-type for bvlan and svlan is 0x88A8 for the Brocade NetIron CES Series and 0x8100 for Brocade NetIron MLX Series devices. The BVLAN, SVLAN, or CVLAN cannot be configured separately on the Brocade NetIron MLX Series device as is done on the Brocade NetIron CES Series.

Application of a standalone ESI

You can use a standalone ESI to perform VLAN ID translation. For example:

```
device(config)# vlan 5
device(config-vlan-5)#tag eth 1/1
device(config-vlan-5)#exit
device(config)#vlan 6
device(config-vlan-6)#tag eth 1/2
device(config)#vlan 7
device(config-vlan-7)#tag eth 1/3
```

Flood domain and VLAN translation

An ESI consisting of VLANs, can optionally be set up as a flood domain, serving two purposes:

- **Flooding** - Creates a domain where packets received on a port within the flood domain are sent to all other ports in the group with proper VLAN translations.
- **VLAN translation -** This feature can be used for translating between SVLANs across a provider boundary.

NOTE

While the flood domain includes multiple VLANs, spanning tree still works within the scope of each VLAN separately. Therefore, loops spanning across VLANs will not get resolved.

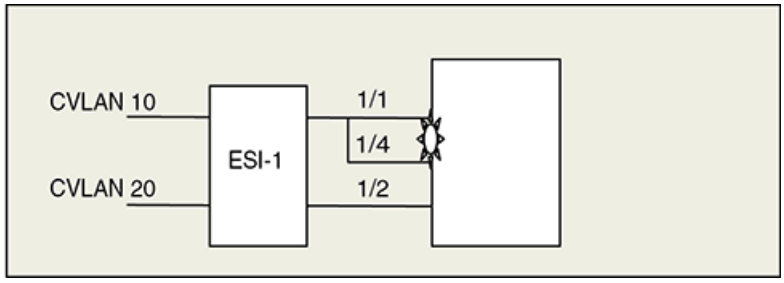
NOTE

Multiple VLANs on same port cannot be added in a single flood domain ESI. However multiple VLANs on different ports are allowed to be added.

System operation without flood domain

Figure 20 shows an ESI configuration with a single flood domain.

FIGURE 19 Single flood domain ESI



The following CLI commands create the scenario shown in Figure 20.

```
device(config)# esi ESI_1 encapsulation cvlan
device(config-esi-ESI_1)# vlan 10
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/1
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/4
device(config-esi-ESI_1-vlan-10)# vlan 20
device(config-esi-ESI_1-vlan-20)# tagged ethernet 1/2
```

System operation without a single flood domain

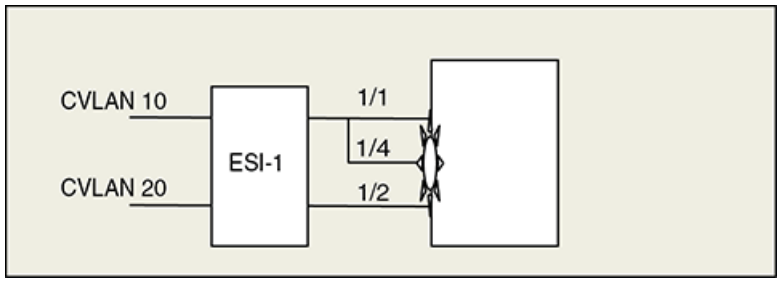
Without a single flood domain configuration, the system operates in the following manner:

- A packet received on 1/1 is sent out on 1/2 with a CVLAN mapping of 20.
- A packet received on 1/2 is sent out on 1/4 with a CVLAN mapping of 10.
- A packet received on 1/4 with a CVLAN mapping of 10 is sent to 1/2 with a CVLAN mapping of 20.

Configuring a flood domain with VLAN translation

You can create a flood domain inside an ESI using the **single-flood-domain** command. A VLAN within an ESI normally defines a flood domain, but when single flood domain is configured, all the VLANs in that ESI become part of one flood domain. In this case every broadcast packet or unknown unicast packet is flooded in all the VLANs in the ESI. When a A C-ESI or S-ESI are configured for single flood domain, they cannot be coupled together.

FIGURE 20 Flood domain



CVLAN translation

Configuring a flood domain with VLAN translation on page 231 shows a configuration for a C-ESI. This combines VLAN 10 and VLAN 20 into one flooding domain.

With this configuration:

- Packets received on 1/1 are sent out on 1/2 with a CVLAN mapping of 20.
- Packets received on 1/2 are sent out on 1/4 with a CVLAN mapping of 10.
- Packets received on 1/4 with a CVLAN mapping of 10 are sent to 1/2 with a CVLAN mapping of 20.

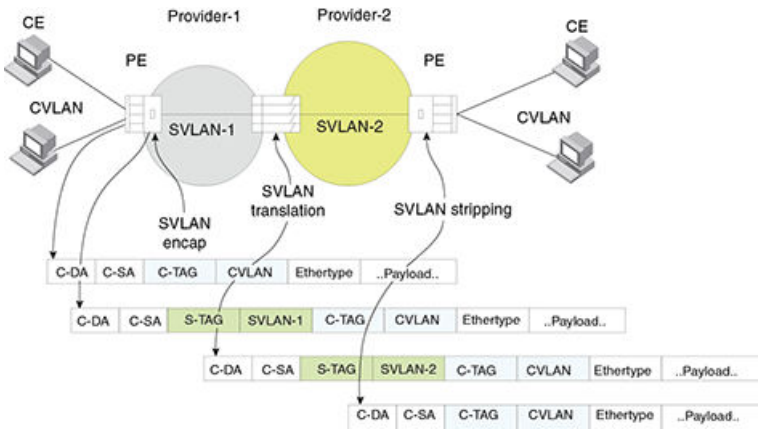
```
device(config)# esi ESI_1 encapsulation cvlan
device(config-esi-ESI_1)# single flood domain
device(config-esi-ESI_1)# vlan 10
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/1
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/4
device(config-esi-ESI_1)# vlan 20
device(config-esi-ESI_1-vlan-20)# tagged ethernet 1/2
```

About IEEE 802.1ad

In a Provider Bridge (PB) network, a provider VLAN is called a Service VLAN (SVLAN), and a customer VLAN is called a Customer VLAN (CVLAN). A CVLAN carries a default tag-type of 0x8100. The range of customer VLANs (CVLANs) can be mapped to an SVLAN, allowing a CVLAN to cross a provider boundary. The SVLAN can be configured to provide service, tunnels, or broadcast domains. The SVLAN and the CVLAN are sent in the same packet so that customer packets with VLAN information are carried to the customer network on the other side.

A Provider Edge (PE) device receives packets with no tags, or packets with CVLAN information, and adds an SVLAN field on the packet before sending to the provider network. The device can be configured to perform SVLAN translation at an inter-provider boundary. Figure 22 provides an example of a PB network.

FIGURE 21 IEEE 802.1ad network



The CVLAN carries a default tag-type of 0x8100. SVLAN encapsulation is similar to CVLAN but with a different tag type (default 0x88a8). A customer's 4K CVLAN domain can be mapped to an SVLAN, allowing the customer VLAN domain to cross a provider boundary. The SVLAN can be configured to provide services, tunnels or broadcast domains.

At an inter-provider boundary, if necessary, the SVLAN value inserted by the first provider may be replaced by a different SVLAN value (this is referred to as SVLAN translation).

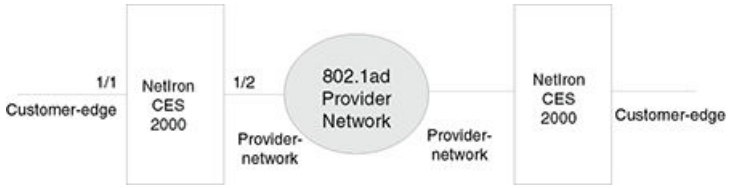
IEEE 802.1ad Provider Bridging limitations

The following provider bridging limitations apply to PB networks:

- An SVLAN can have a value between 1- 4090
- An SVLAN limit of 4K VLANs is typically inadequate in the carrier space.
- As with normal VLAN devices, every PB node must learn all customer MAC addresses, even with SVLAN encapsulation.

Port type configuration for Provider Bridging (PB)

FIGURE 22 Port types in a Provider Bridge (PB)



The Brocade NetIron CES Series defines two types of ports for operation in a provider network:

- **Customer-edge** : This port receives packets with CVLAN tagging. These packets are either switched to other customer-edge ports locally, or are encapsulated with SVLAN tags and are sent out on the provider network ports.
- **Provider-network** : This port receives packets with SVLAN tagging, and transmits packets with SVLAN tagging.

There are two additional port types that are defined for the Brocade NetIron CES Series: these are **backbone-edge** and **backbone-network** . These port types are defined for IEEE 802.1ah Provider Backbone Bridging (PBB).

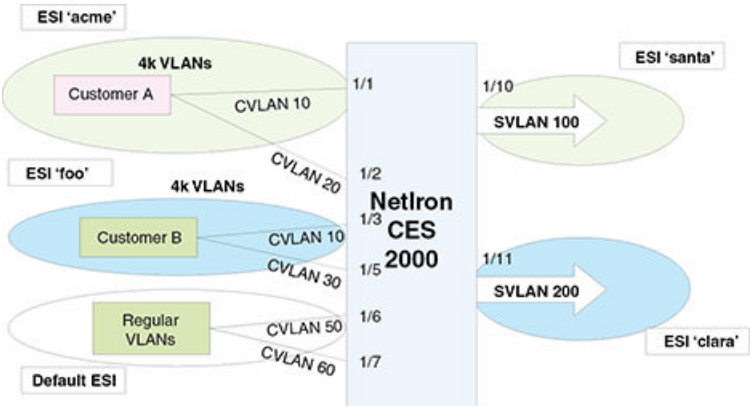
IEEE 802.1ad network configuration example

Configuring a network for IEEE 802.1ad requires the following steps.

1. Configure appropriate port types.
2. Define tag values for CVLAN and SVLAN
3. Define ESIs for CVLAN side and bind VLANs and ports.
4. Define ESIs for SVLAN side and bind VLANs and ports.

Sample configuration

FIGURE 23 IEEE 802.1ad network with ESI definitions



The network architecture for IEEE 802.1ad in Figure 24 shows customer A with two tagged CVLAN ports connected to SVLAN 100, and customer B with two CVLANs connected to SVLAN 200.

To define these configurations and associate them, ESIs are created for each of the configurations. For example, configurations for customer 'A' are defined inside an ESI 'acme' and the carrier-side encapsulation with SVLAN 100 are defined inside an ESI 'santa'.

Once the two ESIs are defined, ESI 'acme' is associated to 'santa' by specifying 'acme' as a Client ESI inside the ESI 'santa'. A similar operation is done for associating customer-side ESI 'foo' with provider-side ESI 'clara' for the customer 'B'.

Configuration steps

The following steps show an example on how to configure an IEEE802.1ad network.

Configure port types for interfaces

Before CVLAN or SVLANs can be provisioned for an interface, the port-type for the interface must be appropriately defined.

The **port-type** command defines a port type for an Ethernet interface. The port-types specify both sides of IEEE 802.1ad and IEEE 802.1ah networks. Enter a command such as the following to set 1/10 and 1/11 to the provider-network port type.

```
device(config)# interface ethernet 1/10
device(config-if-e10000-1/10)# port-type provider-network
device(config-if-e10000-1/10)# exit
```

Syntax: [no] port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the backbone edge port for IEEE 802.1ah PBB.

Use the **backbone-network** parameter to specify the backbone network port for IEEE 802.1ah PBB.

Use the **customer-edge** parameter to specify the customer edge port for IEEE 802.1ad PB.

Use the **provider-network** parameter to specify the provider network port IEEE 802.1ad PB.

Configuring port type values

Four port-type values are specified. For our example, set 1/10 and 1/11 to provider-network port type.

```
device(config)# interface ethernet 1/10
device(config-if-e10000-1/10)# port-type provider-network
device(config-if-e10000-1/10)# exit
```

Use the following commands to set 1/11 to provider-network port type.

```
device(config)#interface ethernet 1/11
device(config-if-e10000-1/11)# port-type provider-network
device(config-if-e10000-1/11)# exit
```

Use the following commands to set 1/1 to customer-edge port type.

```
device(config)#interface ethernet 1/1
device(config-if-e10000-1/1)# port-type customer-edge
device(config-if-e10000-1/1)# exit
```

Syntax: [no] port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the backbone edge port for IEEE 802.1ah PBB.

Use the **backbone-network** parameter to specify the backbone network port for IEEE 802.1ah PBB.

Use the **customer-edge** parameter to specify the customer edge port for IEEE 802.1ad PB.

Use the **provider-network** parameter to specify the provider network port IEEE 802.1ad PB.

Displaying the port type

The **show interfaces** command displays port-type for an interface, as shown below.

```
device(config-if-e10000-1/2)# show interfaces ethernet 1/10
GigabitEthernet1/10 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 0e04.80de.ada0 (bia 0e04.80de.ada0)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of VLAN 4096 (untagged), port is in untagged mode, port state is Disabled
  STP configured to ON, Priority is level0, flow control enabled
  arp-inspection-trust configured to OFF
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  Port-type (802.1ad/802.1ah): provider-network
  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
  NP received 0 packets, Sent to TM 0 packets
  NP Ingress dropped 0 packets
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions
  NP transmitted 0 packets, Received from TM 0 packets
```

TABLE 41 show interfaces command output

This field...	Displays...
Module type Port# is *State*	The *module type* variable specifies a type of interface module, such as 10GigabitEthernet. The *port#* variable specifies the port number for the interface module. The *state* variable if the interface module is up or down.

TABLE 41 show interfaces command output (continued)

This field...	Displays...
Line protocol is *status*	The *status* variable specifies if the line protocol is up or down. If the interface is down due to Link Fault Signaling - Remote Fault Notification (LFS or RFN) the reason is indicated as: "(remote fault)".
STP Root Guard is *status*	The *status* variable specifies if the STP Root Guard is enabled or disabled.
STP BPDU Guard is *status*	The *status* variable specifies if the STP BPDU Guard is enabled or disabled.
Hardware is *module type*	The *module type* variable specifies a type of interface module, such as # Gigabit Ethernet.
Address is *MAC- address*	The *MAC- address* variable specifies the MAC address of the port.
Configured speed and actual speed	The speed that the module has been configured to operate at, and the actual speed it is currently operating at.
Configured port speed and actual duplex	The port capacity that the module has been configured to operate at, and the actual speed it is currently operating at.
Member of *VLAN #* (untagged) *port#* L2 VLANS (tagged) Port is in dual mode/untagged/tagged mode Port state is *status*	The *VLAN#* (untagged) variable specifies a port that is a member of only 1 VLAN. The *port#* L2 VLANS (tagged) variable specifies a port that is a member of multiple ports and untagged. A port is in *dual- mode* specifies member VLAN ports as untagged and tagged. The default mode is dual-mode. The *status* variable identifies the flow of traffic as forwarding or disabled.
STP configured to *status* Priority level Flow control *status*	The *status* variable specifies if the STP is ON or OFF. The priority level assigned to the port-based VLAN. The priority level is on scale from 0-7. The default is 0. The *status* variable is enabled or disabled.
Priority force *status*	The *status* variable specifies if the priority force on a port is disabled on enabled.
Drop precedence level *value*	Identifies the TOS or DSCP value in the IPv4 or IPv6 packet header. The *value* variable specifies the drop precedence on a scale from 0-3. Packets that contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.
Drop precedence force *status*	The *status* variable specifies the drop precedence force as enabled or disabled. Identifies the drop precedence if the force command is configured for a specific ingress port.
arp-inspection-trust configured to *status*	The *status* variable specifies if arp-inspection-trust feature is configured ON or OFF. The default trust setting for a port is untrusted.
Mirror *status*	The *status* variable specifies if the port mirror command is configured as enabled or disabled.
Monitor *status*	The *status* variable specifies if the port monitor command is configured as enabled or disabled.
Trunk membership	The *Trunk membership* variable identifies the interface module as a member of a primary or secondary port. This specifies members of an active port or not a member of an active port.
Configured trunk membership	The *Configured trunk membership* variable identifies the interface module as a member of any configured trunk or not a member of a configured trunk.
Port name	The *port name* variable identifies the name assigned to the port.

TABLE 41 show interfaces command output (continued)

This field...	Displays...
MTU *# bytes* , encapsulation ethernet	Maximum Transmission Unit (MTU) refers to the size of the largest packet or frame that a known layer can pass forward. The *# bytes* variable refers to size of the packet or frame.
#seconds input rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* input rate refers to: • The *value* of bits received per second. • The *value* of packets received per second. • The *%* utilization specifies the port's bandwidth used by received traffic.
# seconds output rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* output rate refers to: • The *value* of bits transmitted per second. • The *value* of packets transmitted per second. • The *%* utilization specifies the port's bandwidth used by transmitted traffic.
value packets input, *value* bytes	• The *value* variable specifies the number of packets received. • The *value* variable specifies the number of bytes received.
Received *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module is receiving on broadcasts, multicasts, and unicast traffic.
value input errors, *value* CRC, *value* frame, *value* ignored	• The *value* variable specifies the number of received packets with errors. • The *value* variable specifies the number of received packets with CRC errors. • The *value* variable specifies the number of received packets with alignment errors. • The *value* variable specifies the number of received packets that are discarded.
value runts, *value* giants	The *value* runts variable specifies the number of small packets that are less than 64 bytes. The *value* giants variable specifies the number of large packets greater than 1518 bytes.
NP received	The number of packets received on the Network Processor (NP).
NP transmitted	The number of packets sent from the Network Processor to the Traffic Manager (TM).
NP Ingress dropped	The number of ingress packets dropped on the Network Processor.
value packets output *value* bytes	• The *value* variable specifies the number of transmitted packets. • The *value* variable specifies the number of transmitted bytes.
Transmitted *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module transmitted on broadcasts, multicasts, and unicast traffic.
value output errors, *value* collisions	• The *value* variable specifies the number of transmitted packets with errors. • The *value* variable specifies the number of transmitted packets with collision errors.
Network Processor transmitted *value* packets Received from Traffic Manager *value* packets	The *value* variable specifies the number of packets transmitted from the Network Processor. The *value* variable specifies the number of packets received by the Network Processor from the Traffic Manager.

ESIs that are associated to a provider ESI are called client ESIs, and packet encapsulation order follows from client ESI to provider ESI.

The ESI concept as defined here can be used for defining and associating all types of services such as IEEE 802.1ad and IEEE 802.1ah.

Creating an ESI

An ESI is configured by giving a name for the ESI and specifying encapsulation type for all the VLANs inside it.

Creating an ESI with CVLAN tagging

To create an ESI with CVLAN tagging named acme, enter a command such as the following.

```
device(config)# esi acme encapsulation cvlan
```

Syntax: [no] esi *esi-name* **encapsulation cvlan** | **svlan** | **isid** | **bvlan**

Use the **cvlan** parameter to specify the encapsulated Customer VLAN (CVLAN).

Use the **svlan** parameter to specify the encapsulated Service VLAN (SVLAN).

Use the **isid** parameter to specify the encapsulated the mapping of different SVLANs into service identifiers.

Use the **bvlan** parameter to specify the encapsulated Backbone VLAN (BVLAN).

Once an ESI is created, subsequent invocations of the ESI do not require encapsulation parameter.

Steps for provisioning a PB network

1. Create ESI for the customer side.
2. Create an ESI and define one or more CVLANs inside it. In the following command acme is the ESI name.

```
device(config)# esi acme encapsulation cvlan
```

3. Define the CVLANs inside the ESI.

```
device(config-esi-acme)# vlan 10
```

4. In the following command, CVLAN 10 becomes tagged on port 1/1

```
device(config-esi-acme-vlan-10)# tagged ethernet 1/1
```

5. In the following command, CVLAN 20 becomes tagged on port 1/2

```
device(config-esi-acme)# vlan 20
device(config-esi-acme-vlan-20)# tagged ethernet 1/2
device(config-esi-acme-vlan-20)# exit
```

Create an ESI on provider side

1. Configure santa as the name of the provider IEEE 802.1ad service

```
device(config)# esi santa encapsulation svlan
```

2. Define SVLAN 100 inside this ESI

```
device(config-esi-acme-iptv)# vlan 100
```

3. Associate physical ports to the VLAN

```
device (config-esi-acme-iptv-vlan-100)# tagged ethernet 1/10
```

Display ESI configuration

At this point, all ESIs have been set up. Enter the **show esi** command to display ESIs.

```
device(config)# show esi
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
acme            cvlan    2                                           0
foo             cvlan    2                                           0
santa           svlan    1                                           0
clara           svlan    1                                           0
```

In this display, ESIs are shown with their encapsulations and number of members (VLANs) in each ESI. At this stage, there are no client-provider bindings.

Associate customer ESI with provider ESI

Now that ESIs are defined for both the customer and provider sides, you can bind the customer ESI to the provider ESI using the **esi-client** command for the ESI named acme.

To complete the configuration for a IEEE 802.1ad network as shown in Sample configuration on page 234, associate the customer ESI to the provider ESI.

```
device (config)# esi santa
device (config-esi-santa)# esi-client acme
device(config-esi-santa)# exit
```

Show ESI command for the final configuration

Enter the **show esi** command to display the final configuration.

```
device(config)# show esi
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
acme            cvlan    2          santa        svlan      100        0
foo             cvlan    2          clara        svlan      200        0
santa           svlan    1                                           1
clara           svlan    1                                           1
```

In this display, the ESI named acme is an ESI of the encapsulation type CVLAN, which has two members (VLANs). The acme ESI now has the santa ESI as a provider with an SVLAN encapsulation type. The santa provider ESI is configured with VLAN ID 100.

This means that both CVLANs in the acme ESI receive a second SVLAN encapsulation (with a tag-type value of SVLAN as globally configured) and a SVLAN ID of 100.

PB using untagged members

For the configuration shown in Sample configuration on page 234, you added a Customer ESI and a Provider ESI and bound them together to provide PB service. This generic configuration can be used when a particular port is shared between customers. However, if a port is completely owned by one customer, you can use the following simple configuration to provide PB service.

Sample configuration 2

1. In this case, customer port 1/1 maps the customer's 4K VLANs. The provider side port 1/2 uses SVLAN 100. All customer traffic *from* the provider side is appended with an SVLAN 100 tag, and for all traffic going *to* the provider side will have the SVLAN 100 tag removed.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# port-type provider-network
```

2. Configure customer ports as provider network ports instead of customer edge ports.

```
device(config)# interface ethernet 1/2
device (config-if-e1000-1/1)# port-type provider-network
```

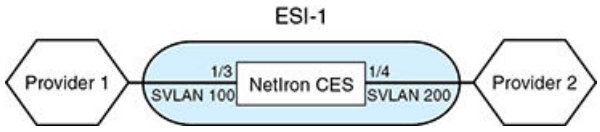
3. Add customer ports as untagged ports so that any traffic tagged with the 8100 tag will also be treated as untagged and will be appended with SVLAN 100.

```
device(config)# esi ESI_1
device(config)# esi ESI_1 encapsulation svlan
device(config-esi-ESI_1)# vlan 100
device(config-esi-ESI_1)# tagged ethernet 1/2
device(config-esi-ESI_1)# untagged ethernet 1/1
device(config-esi-ESI_1)# exit
```

SVLAN translation using flood domain configuration

The **single-flood-domain** command is used for SVLAN translation across provider domains.

FIGURE 24 SVLAN translation at an inter-provider boundary



In the configuration below, packets on port 1/3 with SVLAN=100 are translated to the port 1/4 with SVLAN=200 as the ports are in the same flood domain.

```
device(config)# esi ESI_1 encapsulation svlan
device(config-esi-ESI_1)# single-flood-domain
device(config-esi-ESI_1)# vlan 100
device(config-esi-ESI_1)# tagged ethernet 1/3
device(config-esi-ESI_1)# vlan 200
device(config-esi-ESI_1)# tagged ethernet 1/4
device(config-esi-ESI_1)# exit
```

Untagged ports

Packets from SVLAN port with only SVLAN tag (no CVLAN tag) are sent to only untagged ports in the default ESI.

Default VLAN

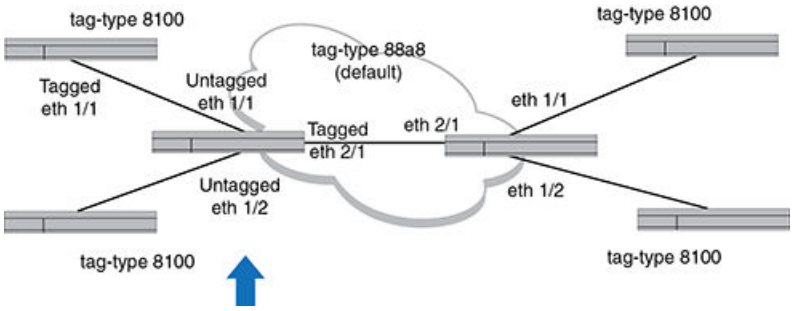
Packets from SVLAN with default VLAN as CVLAN tag are sent to untagged ports in the default ESI.

No Default VLANs are allowed inside non-default ESIs.

Port-based Service Interface Super AggregatedVLANs (SAV)

Using Port-based Service Interfaces is equivalent to using SAV in other Brocade products. Port-based Service Interfaces can be used when mapping a port based interface to a single Service VLAN Tag (S-TAG). When using a port-based service interface, it maps all VLAN-IDs from incoming ports to a SVLAN as shown in Port-based Service Interface Super AggregatedVLANs (SAV).

Port-based service interface



Sample configuration

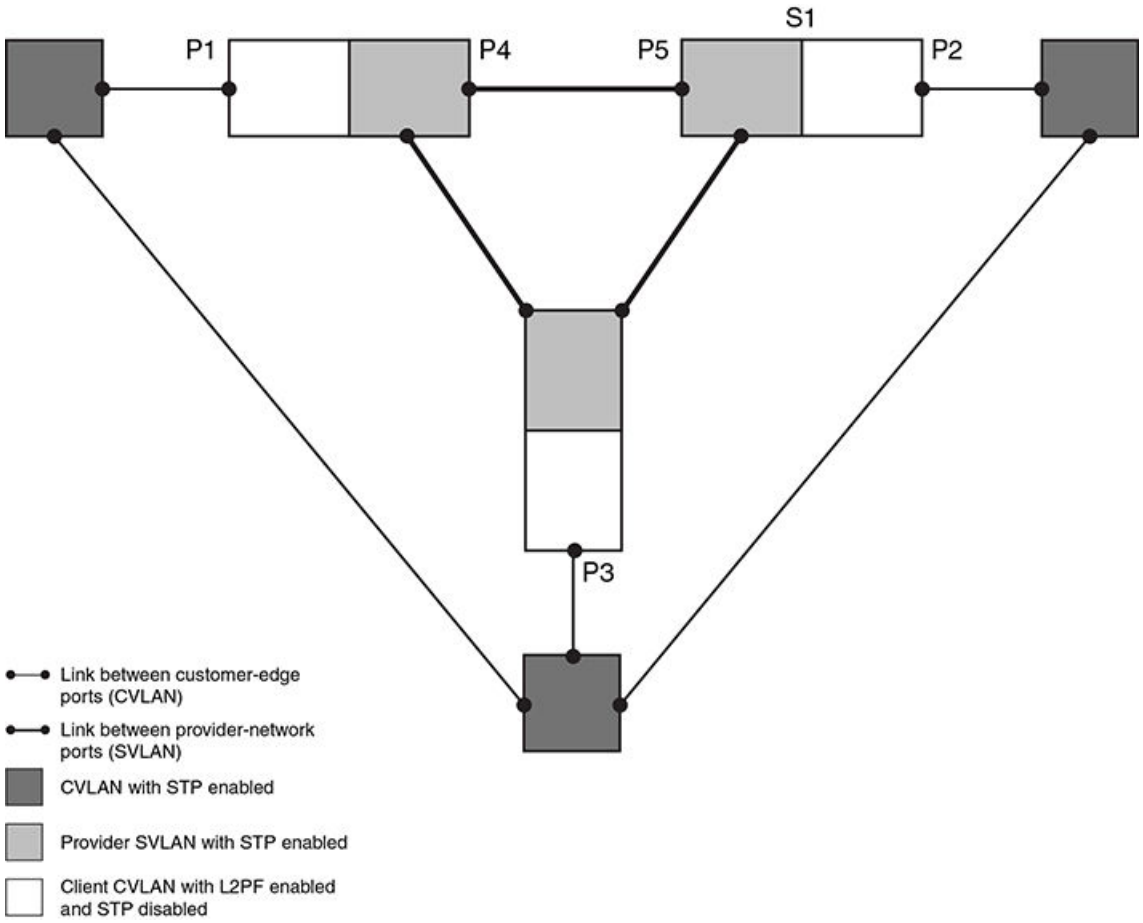
```
device(config)# tag-type tag2 ethernet 2/1
device(config)# vlan 100
device(config-vlan-100)# tagged ethernet 2/1
device(config-vlan-100)# untagged ethernet 1/1
device(config)# vlan 200
device(config-vlan-200)# tagged ethernet 2/1
device(config-vlan-200)#untagged ethernet 1/2
```

Layer 2 Protocol Forwarding (L2PF)

Layer 2 Protocol Forwarding (L2PF) is configured on CVLANs. You can configure the system to forward BPDUs on all CVLANs coming at different edge ports, drop all BPDUs, or selectively enable forwarding on a few CVLANs.

L2PF can transparently extend the STP topology of the CVLAN domain through the SVLAN domain, as if the CVLAN switches were directly hooked together. L2PF can be applied to CVLAN, which is a client of provider SVLAN as shown in "L2PF in a network".

FIGURE 25 L2PF in a network

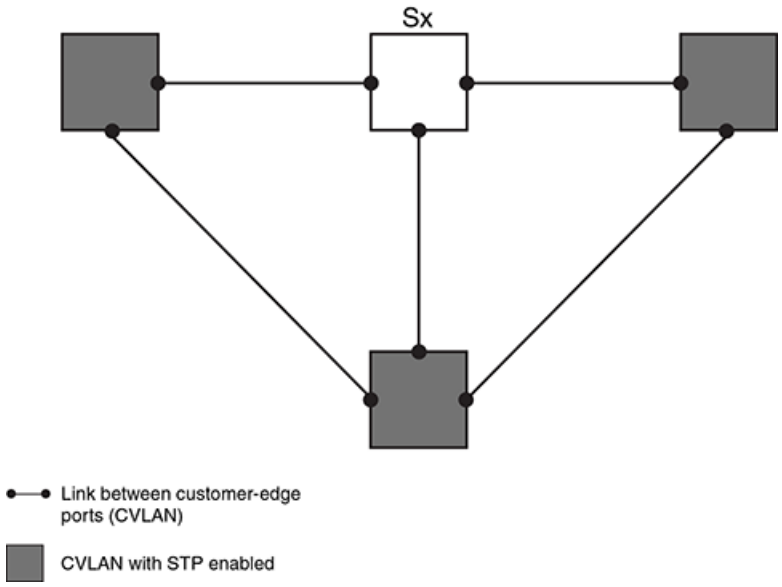


In this network, a CVLAN BPDU ingressing port P1 gets encapsulated and forwarded through the SVLAN domain as an ordinary multicast frame and then it egresses ports P2 and P3. In the SVLAN domain, a SVLAN BPDU originating from P4 and the ingressing port P5 gets terminated and processed in switch S1 for STP calculation. In other words, the final STP topology in the SVLAN domain is completely independent from the final STP topology in the CVLAN domain.

From the CVLAN domain's point of view, the physical network appears as shown in Figure 27.

Below, Sx is a simple bridge without STP.

FIGURE 26 Physical network



NOTE
It is critical for L2PF to disable STP on the client CVLAN to operate in that CVLAN.

STP wins over L2PF when both are enabled on a given client CVLAN. Table 42 displays the Port configuration for IEEE 802.1ah and IEEE 802.1ad.

TABLE 42 Port Configuration

Description	L2PF	STPF
CVLAN BPDU flooded inside CVLAN just like any multicast frame (but not flooded to SVLAN)	Disabled	Disabled
CVLAN BPDU terminated and processed in client CVLAN	Disabled	Enabled
CVLAN BPDU tunneled to provider SVLAN, and flooded in provider SVLAN	Enabled	Disabled
CVLAN BPDU terminated and processed in client CVLAN	Enabled	Enabled

NOTE
The behavior of L2PF is independent on whether the STP in the provider SVLAN is enabled or not.

Global configuration

By default, the device enables Layer 2 protocol forwarding (L2PF). Customer-BPDU packets on all CVLANs are forwarded to provider network ports when they arrive at customer-edge ports.

Since L2PF is globally enabled by default, all CVLANs have L2PF enabled by default. No CLI configuration is needed.

To globally disable Layer 2 protocol forwarding, enter the following command:

device(config)# no l2protocol-forwarding stp

To disable L2PF on a particular CVLAN, enter commands such as the following:

```
device(config)#esi acme encapsulation cvlan
device(config-abc)#vlan 10
device(config-abc-vlan-10)# no l2protocol-forward stp
```

Syntax: [no] l2protocol-forwarding stp

Use the **no** parameter to disable Layer 2 protocol forwarding.

Ethernet Service Instance for Brocade NetIron CES Series and Brocade NetIron CER Series Devices

ESI overview

An Ethernet Service Instance (ESI) is a provisioning environment for defining VLAN and other layer 2 parameters for creating services, typically across a carrier network.

In a local area network a total of 4K VLANs can be configured across the entire network domain. With a Q-in-Q bridging, VLANs from the set of 4K VLANs can be inter-connected across a provider network. While ESI allows a carrier to provide transport services for different sets of 4K VLANs for different customers, the provider network is still limited to using 4K VLANs across all of the customers connected to a single box, as it is very difficult to configure and manage different sets of 4K VLANs across the different ports within a single system.

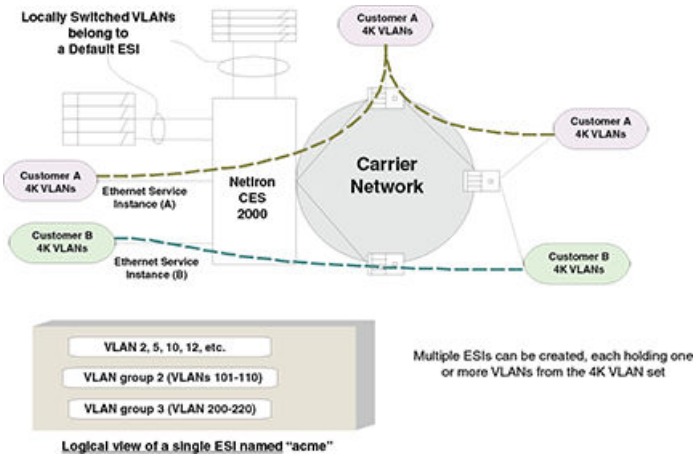
Using an ESI, a carrier can create service instances that hold one or more VLANs. Each instance has an alphanumeric name that is locally unique. The purpose of creating an instance is to provide a container to hold VLANs and other layer2 parameters that define properties of all of the elements contained within the instance.

VLANs are added to an ESI using a standard VLAN command for individual VLANs or a VLAN group command for adding a set of VLANs.

In a simplified network shown in ESI overview on page 225, customers A and B are connected to a Brocade NetIron CES Series device with each customer having a separate set of 4K VLANs. One or more ESIs are created to hold these 4K VLANs.

> NOTE
> Although theoretically it is possible to add sets of 4K VLANs in ESIs, the actual number of VLANs in an ESI is limited by the use of memory and hardware resources.

Ethernet Service Instance for VLAN configuration



Once an ESI is defined, Brocade NetIron CES Series and Brocade NetIron CER Series devices operate on rules for configuring VLANs inside an ESI, and check against configuration incompatibilities (such as configuring the same VLAN value from two different ESIs on the same port).

Types of ESI

There are two types of Ethernet Service Instances, as described:

Default ESI

In ESI overview on page 225, VLANs associated with ports in the top left corner of the Brocade NetIron CES Series and Brocade NetIron CER Series devices aren't being transported over to the carrier network - these VLANs are being locally switched and connected with switches in the local area network. The Brocade NetIron CES Series and Brocade NetIron CER Series devices support 4K VLANs of this type, without any ESI configured. Internally, these VLANs are associated with a Default ESI, and are referred to as 'Regular VLANs'.

Customer ESI

ESIs that are configured to hold customer VLANs that need to be transported across a carrier network are usually referred to as customer ESIs. A customer ESI always has a Layer 2 protocol VLAN encapsulation applied to all VLANs in the ESI.

In a carrier network, an incoming customer VLAN packet will usually be configured with successive encapsulations, such as with service VLANs, in-service identifiers, or backbone VLANs. Each encapsulation is associated with a different ESI. To define the encapsulation hierarchy, an ESI for an incoming packet is defined as a client of the ESI for the next encapsulation. The next encapsulation ESI is referred to as a 'provider ESI' and the ESIs that are declared as client ESIs are referred to as 'client ESIs'.

Depending on their association, customer ESIs can be one of the three types:

- Standalone ESI - An ESI that is not linked to any other ESI, and are used only to hold VLANs and define their properties.
- Provider ESI - An ESI with one VLAN, and one or more client ESIs, each holding one or more VLANs.
- Client ESI - An ESI that is defined to be a client of another ESI, and that can have one or more VLANs defined inside it.

Configuration considerations

The following rules apply for CLI operations for Provider Bridge (PB) and related protocols:

- To prevent topology changes at startup, it is recommended that you not use the same ESI-Vlan ID as the Default-Vlan ID.
- An ESI is created for each service (such as a customer CVLANs, SVLANs, PBB, etc.).
- All attributes for an ESI - such as VLAN, port binding, encapsulation, etc., are defined inside an ESI.
- To prevent configuration errors, no parameter overrides are permitted outside of an ESI.
- ESIs can be nested to provide multiple protocol encapsulations for a packet. Restrictions on bindings can be present depending on the actual platform. When nesting is used, inner ESIs are called client ESIs.
- A given VLAN means CVLAN or SVLAN, depending on the encapsulation definition for the ESI:
  - If no encapsulation is defined as "cvlan" the VLAN refers to CVLAN.

    NOTE
    For ESI VLANs, It is mandatory to define an encapsulation.

-
  - If encapsulation is "svlan", the VLAN refers to SVLAN (PB).
  - If encapsulation is "bvlan", the VLAN refers to B-VLAN (PBB).
- ISID values are treated differently from other VLANs, since ISID parameters have no networking association and are used only for mapping different SVLANs into service identifiers.

Creating an ESI

Create an ESI by naming it and specifying encapsulation type for all the VLANs inside it.

Creating an ESI with CVLAN tagging

To create an ESI with CVLAN tagging named "acme", enter a command such as the following.

```
device (config)# esi acme encapsulation cvlan
```

Syntax: [no] esi *esi-name* **encapsulation cvlan** | **svlan** | **isid** | **bvlan**

Use the **cvlan** parameter to specify the encapsulated customer VLAN (CVLAN).

Use the **svlan** parameter to specify the encapsulated service VLAN (SVLAN).

Use the **isid** parameter to specify the encapsulation for the mapping of SVLANs into service identifiers.

Use the **bvlan** parameter to specify the encapsulated backbone VLAN (BVLAN).

Once an ESI is created, subsequent invocations of the ESI do not require the encapsulation parameter.

Defining CVLANs inside the ESI

To define CVLANs inside the ESI, enter a command such as the following.

```
device(config-esi-acme)# vlan 10
```

Configuring the CVLAN to be tagged

To configure CVLAN 10 to be tagged on port 1/1, enter a command such as the following.

```
device(config-esi-acme-vlan-10)# tagged ethernet 1/1
```

Show VLAN commands

The following **show** commands will display custom ESI configurations for VLANs.

Displaying information for a VLAN inside an ESI

To display a VLAN inside an ESI, enter a command such as the following.

```
device(config)#show esi acme vlan 10
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
VLAN 10 details:
----------------
PORT-VLAN 10, Name [None], Priority Level-,Priority Force 0
L2 protocols   : NONE
ESI: bay Encapsulation: cvlan
----------------
No ports associated with VLAN
Arp Inspection: 0
DHCP Snooping: 0
L2 protocol forwarding mode:Tunnel
Flood domain ID 4176
```

Syntax: show vlan *num*

Displaying information for a VLAN inside an ESI in brief format

The **show vlan brief** command displays VLANs in a tabular format for compactness. This command may be executed from any CLI level.

```
device#show vlan brief
Configured PORT-VLAN entries: 1
Maximum PORT-VLAN entries: 512
Default PORT-VLAN id: 1
VLAN     Name       Encap ESI                              Pri Ports
----     ----       ----- ---                              --- -----
10       [None]     cvlan acme                             -
```

Syntax: show vlan brief

Displaying a single ESI

To display details of a single ESI, enter the following command from any level of the CLI.

```
device#show esi acme
ESI Name        Encap    Number of  Provider        Provider   Provider Client
                         Members    ESI             Encap      VLAN     ESIs
--------        -----    ---------  --------        -----      -------- ------
acme            cvlan    1                                           0
VLAN(s) at this ESI:
--------------------
VLAN     Name       Pri [L2 Protocols]       Ports
10       [None]     cvlan acme                             -   NONE
```

Displaying all ESIs

Use the **show esi** command to display a list of all the ESIs configured in the system. This command can be used at any level of the CLI.

```
device(config)#show esi
ESI Name        Encap    Number of  Provider        Provider   Provider Client
                         Members    ESI             Encap      VLAN     ESIs
--------        -----    ---------  --------        -----      -------- ------
acme            cvlan    1                                           0
```

Syntax: show esi *name*

Tag-type configuration

For the Brocade NetIron CES Series and Brocade NetIron CER Series, the following two VLAN tag-types are allowed that can be configured globally:

- **tag1** applies to customer edge ports (CVLAN) by default.
- **tag2** applies to provider-network, backbone-edge, and backbone-network port types (SVLAN and BVLAN) by default.

  NOTE
  The **tag1** and **tag2** are independent of port-types, so the system can be configured to use **tag1** for SVLAN, BVLAN and **tag2** for CVLAN.

Configuring tag-types

You can set the ISID value using a separate command similar to the Brocade NetIron XMR Series and Brocade NetIron MLX Series command as shown below.

Syntax: [no] tag-value isid *num*

You can configure CVLAN, SVLAN, and BVLAN tag-types as shown below.

```
device(config)# tag-value tag1 8100
device(config)# tag-value tag2 9100
device(config)# tag-type cvlan tag1 svlan tag2 bvlan tag2
```

Syntax: [no] tag-value *num*

Syntax: tag-type *tag-n*

The *num* parameter specifies the value assigned to the tag. The default value for **tag1** is 0x8100 and for **tag2** is 0x88a8.

The *tag-n* parameter can be either **tag1** or **tag2** .

Tag type can be changed from a default value to a specific port as shown in the following example.

```
device(config-if-e1000-1/1)# tag-type tag2 ethernet 1/1
device(config-if-e1000-1/1)# tag-type tag1 ethernet 1/2
```

Syntax: tag-type *tagid* **ethernet** *interface_id*

The *tagid* parameter can be either **tag1** or **tag2** . Possible tagid values are:

- – isid - to set the isid tag-value
  – tag1 - to set the tag-type tag1 value
  – tag2- to set the tag-type tag2 value

The *interface_id* parameter specifies the Ethernet slot and port ID.

Restrictions

The tag-type has the following restrictions:

- CVLAN and SVLAN cannot have the same tag-type but the tag-value can be set to the same.
- SVLAN and BVLAN must have the same tag-type.
- Port-type must be set to the default to configure the port-level tag-type.

Displaying tag types

To display the different tag types, enter a command such as the following.

```
device(config)#show tag-type
Encap      Current VLAN Tags             Default VLAN Tags
-----      -----------------             -----------------
cvlan                 8100                       8100
svlan                 9100                       88A8
isid                  86B5                       88E7
bvlan                 9100                       88A8
```

NOTE

The Brocade NetIron CES Series and Brocade NetIron CER Series require that the SVLAN and BVLAN tag types be same. The default tag-type for bvlan and svlan is 0x88A8 for the Brocade NetIron CES Series and 0x8100 for Brocade NetIron MLX Series devices. The BVLAN, SVLAN, or CVLAN cannot be configured separately on the Brocade NetIron MLX Series device as is done on the Brocade NetIron CES Series.

Application of a standalone ESI

You can use a standalone ESI to perform VLAN ID translation. For example:

```
device(config)# vlan 5
device(config-vlan-5)#tag eth 1/1
device(config-vlan-5)#exit
device(config)#vlan 6
device(config-vlan-6)#tag eth 1/2
device(config)#vlan 7
device(config-vlan-7)#tag eth 1/3
```

Flood domain and VLAN translation

An ESI consisting of VLANs, can optionally be set up as a flood domain, serving two purposes:

- **Flooding** - Creates a domain where packets received on a port within the flood domain are sent to all other ports in the group with proper VLAN translations.
- **VLAN translation -** This feature can be used for translating between SVLANs across a provider boundary.

NOTE

While the flood domain includes multiple VLANs, spanning tree still works within the scope of each VLAN separately. Therefore, loops spanning across VLANs will not get resolved.

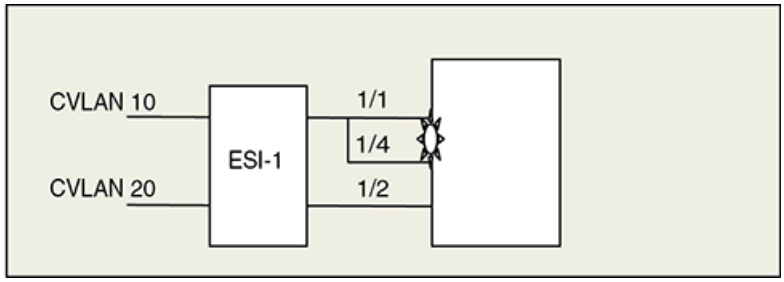
NOTE

Multiple VLANs on same port cannot be added in a single flood domain ESI. However multiple VLANs on different ports are allowed to be added.

System operation without flood domain

Figure 29 shows an ESI configuration with a single flood domain.

FIGURE 27 Single flood domain ESI



The following CLI commands create the scenario shown in Figure 29.

```
device(config)# esi ESI_1 encapsulation cvlan
device(config-esi-ESI_1)# vlan 10
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/1
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/4
device(config-esi-ESI_1-vlan-10)# vlan 20
device(config-esi-ESI_1-vlan-20)# tagged ethernet 1/2
```

System operation without a single flood domain

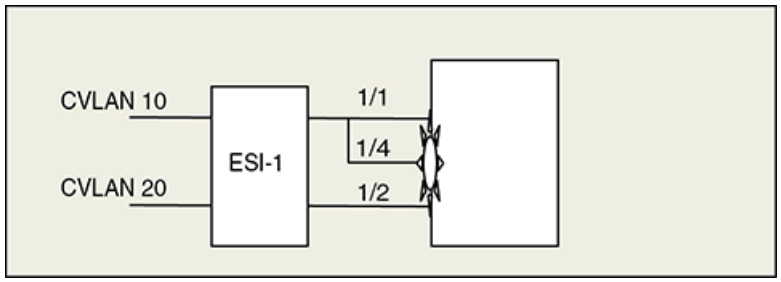
Without a single flood domain configuration, the system operates in the following manner:

- A packet received on 1/1 is sent out on 1/2 with a CVLAN mapping of 20.
- A packet received on 1/2 is sent out on 1/4 with a CVLAN mapping of 10.
- A packet received on 1/4 with a CVLAN mapping of 10 is sent to 1/2 with a CVLAN mapping of 20.

Configuring a flood domain with VLAN translation

You can create a flood domain inside an ESI using the **single-flood-domain** command. A VLAN within an ESI normally defines a flood domain, but when single flood domain is configured, all the VLANs in that ESI become part of one flood domain. In this case every broadcast packet or unknown unicast packet is flooded in all the VLANs in the ESI. When a A C-ESI or S-ESI are configured for single flood domain, they cannot be coupled together.

FIGURE 28 Flood domain



CVLAN translation

Configuring a flood domain with VLAN translation on page 231 shows a configuration for a C-ESI. This combines VLAN 10 and VLAN 20 into one flooding domain.

With this configuration:

- Packets received on 1/1 are sent out on 1/2 with a CVLAN mapping of 20.
- Packets received on 1/2 are sent out on 1/4 with a CVLAN mapping of 10.
- Packets received on 1/4 with a CVLAN mapping of 10 are sent to 1/2 with a CVLAN mapping of 20.

```
device(config)# esi ESI_1 encapsulation cvlan
device(config-esi-ESI_1)# single flood domain
device(config-esi-ESI_1)# vlan 10
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/1
device(config-esi-ESI_1-vlan-10)# tagged ethernet 1/4
device(config-esi-ESI_1)# vlan 20
device(config-esi-ESI_1-vlan-20)# tagged ethernet 1/2
```

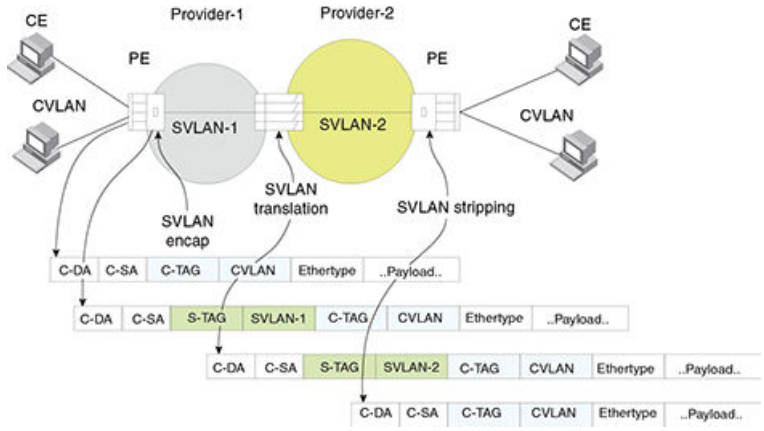
IEEE 802.1ad - Provider Bridges for the Brocade NetIron CES Series and Brocade NetIron CER Series

About IEEE 802.1ad

In a Provider Bridge (PB) network, a provider VLAN is called a Service VLAN (SVLAN), and a customer VLAN is called a Customer VLAN (CVLAN). A CVLAN carries a default tag-type of 0x8100. The range of customer VLANs (CVLANs) can be mapped to an SVLAN, allowing a CVLAN to cross a provider boundary. The SVLAN can be configured to provide service, tunnels, or broadcast domains. The SVLAN and the CVLAN are sent in the same packet so that customer packets with VLAN information are carried to the customer network on the other side.

A Provider Edge (PE) device receives packets with no tags, or packets with CVLAN information, and adds an SVLAN field on the packet before sending to the provider network. The device can be configured to perform SVLAN translation at an inter-provider boundary. Figure 31 provides an example of a PB network.

FIGURE 29 IEEE 802.1ad network



The CVLAN carries a default tag-type of 0x8100. SVLAN encapsulation is similar to CVLAN but with a different tag type (default 0x88a8). A customer's 4K CVLAN domain can be mapped to an SVLAN, allowing the customer VLAN domain to cross a provider boundary. The SVLAN can be configured to provide services, tunnels or broadcast domains.

At an inter-provider boundary, if necessary, the SVLAN value inserted by the first provider may be replaced by a different SVLAN value (this is referred to as SVLAN translation).

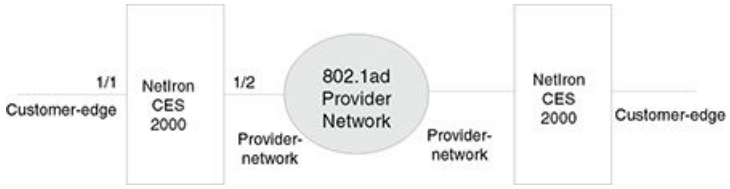
IEEE 802.1ad Provider Bridging limitations

The following provider bridging limitations apply to PB networks:

- An SVLAN can have a value between 1- 4090
- An SVLAN limit of 4K VLANs is typically inadequate in the carrier space.
- As with normal VLAN devices, every PB node must learn all customer MAC addresses, even with SVLAN encapsulation.

Port type configuration for Provider Bridging (PB)

FIGURE 30 Port types in a Provider Bridge (PB)



The Brocade NetIron CES Series defines two types of ports for operation in a provider network:

- **Customer-edge** : This port receives packets with CVLAN tagging. These packets are either switched to other customer-edge ports locally, or are encapsulated with SVLAN tags and are sent out on the provider network ports.
- **Provider-network** : This port receives packets with SVLAN tagging, and transmits packets with SVLAN tagging.

There are two additional port types that are defined for the Brocade NetIron CES Series: these are **backbone-edge** and **backbone-network** . These port types are defined for IEEE 802.1ah Provider Backbone Bridging (PBB).

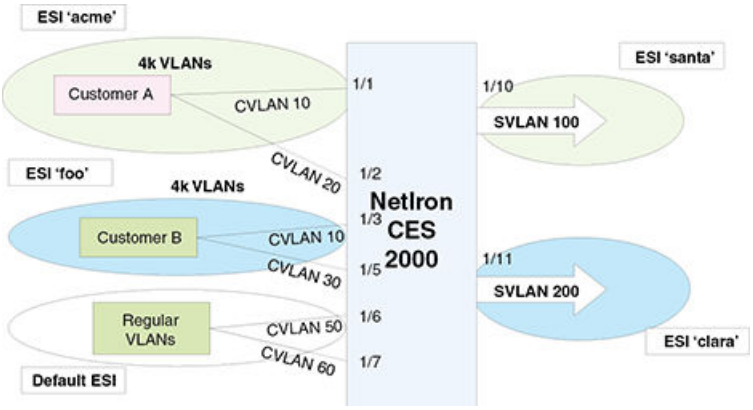
IEEE 802.1ad network configuration example

Configuring a network for IEEE 802.1ad requires the following steps.

1. Configure appropriate port types.
2. Define tag values for CVLAN and SVLAN
3. Define ESIs for CVLAN side and bind VLANs and ports.
4. Define ESIs for SVLAN side and bind VLANs and ports.

Sample configuration

FIGURE 31 IEEE 802.1ad network with ESI definitions



The network architecture for IEEE 802.1ad in Figure 33 shows customer A with two tagged CVLAN ports connected to SVLAN 100, and customer B with two CVLANs connected to SVLAN 200.

To define these configurations and associate them, ESIs are created for each of the configurations. For example, configurations for customer 'A' are defined inside an ESI 'acme' and the carrier-side encapsulation with SVLAN 100 are defined inside an ESI 'santa'.

Once the two ESIs are defined, ESI 'acme' is associated to 'santa' by specifying 'acme' as a Client ESI inside the ESI 'santa'. A similar operation is done for associating customer-side ESI 'foo' with provider-side ESI 'clara' for the customer 'B'.

Configuration steps

The following steps show an example on how to configure an IEEE802.1ad network.

Configure port types for interfaces

Before CVLAN or SVLANs can be provisioned for an interface, the port-type for the interface must be appropriately defined.

The **port-type** command defines a port type for an Ethernet interface. The port-types specify both sides of IEEE 802.1ad and IEEE 802.1ah networks. Enter a command such as the following to set 1/10 and 1/11 to the provider-network port type.

```
device(config)# interface ethernet 1/10
device(config-if-e10000-1/10)# port-type provider-network
device(config-if-e10000-1/10)# exit
```

Syntax: [no] port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the backbone edge port for IEEE 802.1ah PBB.

Use the **backbone-network** parameter to specify the backbone network port for IEEE 802.1ah PBB.

Use the **customer-edge** parameter to specify the customer edge port for IEEE 802.1ad PB.

Use the **provider-network** parameter to specify the provider network port IEEE 802.1ad PB.

Configuring port type values

Four port-type values are specified. For our example, set 1/10 and 1/11 to provider-network port type.

```
device(config)# interface ethernet 1/10
device(config-if-e10000-1/10)# port-type provider-network
device(config-if-e10000-1/10)# exit
```

Use the following commands to set 1/11 to provider-network port type.

```
device(config)#interface ethernet 1/11
device(config-if-e10000-1/11)# port-type provider-network
device(config-if-e10000-1/11)# exit
```

Use the following commands to set 1/1 to customer-edge port type.

```
device(config)#interface ethernet 1/1
device(config-if-e10000-1/1)# port-type customer-edge
device(config-if-e10000-1/1)# exit
```

Syntax: [no] port-type [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the backbone edge port for IEEE 802.1ah PBB.

Use the **backbone-network** parameter to specify the backbone network port for IEEE 802.1ah PBB.

Use the **customer-edge** parameter to specify the customer edge port for IEEE 802.1ad PB.

Use the **provider-network** parameter to specify the provider network port IEEE 802.1ad PB.

Displaying the port type

The **show interfaces** command displays port-type for an interface, as shown below.

```
device(config-if-e10000-1/2)# show interfaces ethernet 1/10
GigabitEthernet1/10 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
  Hardware is GigabitEthernet, address is 0e04.80de.ada0 (bia 0e04.80de.ada0)
  Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
  Member of VLAN 4096 (untagged), port is in untagged mode, port state is Disabled
  STP configured to ON, Priority is level0, flow control enabled
  arp-inspection-trust configured to OFF
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  Port-type (802.1ad/802.1ah): provider-network
  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
  NP received 0 packets, Sent to TM 0 packets
  NP Ingress dropped 0 packets
  0 packets output, 0 bytes, 0 underruns
  Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
  0 output errors, 0 collisions
  NP transmitted 0 packets, Received from TM 0 packets
```

TABLE 43 show interfaces command output

This field...	Displays...
Module type Port# is *State*	The *module type* variable specifies a type of interface module, such as 10GigabitEthernet. The *port#* variable specifies the port number for the interface module. The *state* variable if the interface module is up or down.

TABLE 43 show interfaces command output (continued)

This field...	Displays...
Line protocol is *status*	The *status* variable specifies if the line protocol is up or down. If the interface is down due to Link Fault Signaling - Remote Fault Notification (LFS or RFN) the reason is indicated as: "(remote fault)".
STP Root Guard is *status*	The *status* variable specifies if the STP Root Guard is enabled or disabled.
STP BPDU Guard is *status*	The *status* variable specifies if the STP BPDU Guard is enabled or disabled.
Hardware is *module type*	The *module type* variable specifies a type of interface module, such as # Gigabit Ethernet.
Address is *MAC- address*	The *MAC- address* variable specifies the MAC address of the port.
Configured speed and actual speed	The speed that the module has been configured to operate at, and the actual speed it is currently operating at.
Configured port speed and actual duplex	The port capacity that the module has been configured to operate at, and the actual speed it is currently operating at.
Member of *VLAN #* (untagged) *port#* L2 VLANS (tagged) Port is in dual mode/untagged/tagged mode Port state is *status*	The *VLAN#* (untagged) variable specifies a port that is a member of only 1 VLAN. The *port#* L2 VLANS (tagged) variable specifies a port that is a member of multiple ports and untagged. A port is in *dual- mode* specifies member VLAN ports as untagged and tagged. The default mode is dual-mode. The *status* variable identifies the flow of traffic as forwarding or disabled.
STP configured to *status* Priority level Flow control *status*	The *status* variable specifies if the STP is ON or OFF. The priority level assigned to the port-based VLAN. The priority level is on scale from 0-7. The default is 0. The *status* variable is enabled or disabled.
Priority force *status*	The *status* variable specifies if the priority force on a port is disabled on enabled.
Drop precedence level *value*	Identifies the TOS or DSCP value in the IPv4 or IPv6 packet header. The *value* variable specifies the drop precedence on a scale from 0-3. Packets that contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.
Drop precedence force *status*	The *status* variable specifies the drop precedence force as enabled or disabled. Identifies the drop precedence if the force command is configured for a specific ingress port.
arp-inspection-trust configured to *status*	The *status* variable specifies if arp-inspection-trust feature is configured ON or OFF. The default trust setting for a port is untrusted.
Mirror *status*	The *status* variable specifies if the port mirror command is configured as enabled or disabled.
Monitor *status*	The *status* variable specifies if the port monitor command is configured as enabled or disabled.
Trunk membership	The *Trunk membership* variable identifies the interface module as a member of a primary or secondary port. This specifies members of an active port or not a member of an active port.
Configured trunk membership	The *Configured trunk membership* variable identifies the interface module as a member of any configured trunk or not a member of a configured trunk.
Port name	The *port name* variable identifies the name assigned to the port.

TABLE 43 show interfaces command output (continued)

This field...	Displays...
MTU *# bytes* , encapsulation ethernet	Maximum Transmission Unit (MTU) refers to the size of the largest packet or frame that a known layer can pass forward. The *# bytes* variable refers to size of the packet or frame.
#seconds input rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* input rate refers to: • The *value* of bits received per second. • The *value* of packets received per second. • The *%* utilization specifies the port's bandwidth used by received traffic.
# seconds output rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* output rate refers to: • The *value* of bits transmitted per second. • The *value* of packets transmitted per second. • The *%* utilization specifies the port's bandwidth used by transmitted traffic.
value packets input, *value* bytes	• The *value* variable specifies the number of packets received. • The *value* variable specifies the number of bytes received.
Received *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module is receiving on broadcasts, multicasts, and unicast traffic.
value input errors, *value* CRC, *value* frame, *value* ignored	• The *value* variable specifies the number of received packets with errors. • The *value* variable specifies the number of received packets with CRC errors. • The *value* variable specifies the number of received packets with alignment errors. • The *value* variable specifies the number of received packets that are discarded.
value runts, *value* giants	The *value* runts variable specifies the number of small packets that are less than 64 bytes. The *value* giants variable specifies the number of large packets greater than 1518 bytes.
NP received	The number of packets received on the Network Processor (NP).
NP transmitted	The number of packets sent from the Network Processor to the Traffic Manager (TM).
NP Ingress dropped	The number of ingress packets dropped on the Network Processor.
value packets output *value* bytes	• The *value* variable specifies the number of transmitted packets. • The *value* variable specifies the number of transmitted bytes.
Transmitted *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module transmitted on broadcasts, multicasts, and unicast traffic.
value output errors, *value* collisions	• The *value* variable specifies the number of transmitted packets with errors. • The *value* variable specifies the number of transmitted packets with collision errors.
Network Processor transmitted *value* packets Received from Traffic Manager *value* packets	The *value* variable specifies the number of packets transmitted from the Network Processor. The *value* variable specifies the number of packets received by the Network Processor from the Traffic Manager.

ESIs that are associated to a provider ESI are called client ESIs, and packet encapsulation order follows from client ESI to provider ESI.

The ESI concept as defined here can be used for defining and associating all types of services such as IEEE 802.1ad and IEEE 802.1ah.

Creating an ESI

An ESI is configured by giving a name for the ESI and specifying encapsulation type for all the VLANs inside it.

Creating an ESI with CVLAN tagging

To create an ESI with CVLAN tagging named acme, enter a command such as the following.

```
device(config)# esi acme encapsulation cvlan
```

Syntax: [no] esi *esi-name* **encapsulation cvlan** | **svlan** | **isid** | **bvlan**

Use the **cvlan** parameter to specify the encapsulated Customer VLAN (CVLAN).

Use the **svlan** parameter to specify the encapsulated Service VLAN (SVLAN).

Use the **isid** parameter to specify the encapsulated the mapping of different SVLANs into service identifiers.

Use the **bvlan** parameter to specify the encapsulated Backbone VLAN (BVLAN).

Once an ESI is created, subsequent invocations of the ESI do not require encapsulation parameter.

Steps for provisioning a PB network

1. Create ESI for the customer side.
2. Create an ESI and define one or more CVLANs inside it. In the following command acme is the ESI name.

```
device(config)# esi acme encapsulation cvlan
```

3. Define the CVLANs inside the ESI.

```
device(config-esi-acme)# vlan 10
```

4. In the following command, CVLAN 10 becomes tagged on port 1/1

```
device(config-esi-acme-vlan-10)# tagged ethernet 1/1
```

5. In the following command, CVLAN 20 becomes tagged on port 1/2

```
device(config-esi-acme)# vlan 20
device(config-esi-acme-vlan-20)# tagged ethernet 1/2
device(config-esi-acme-vlan-20)# exit
```

Create an ESI on provider side

1. Configure santa as the name of the provider IEEE 802.1ad service

```
device(config)# esi santa encapsulation svlan
```

2. Define SVLAN 100 inside this ESI

```
device(config-esi-acme-iptv)# vlan 100
```

3. Associate physical ports to the VLAN

```
device (config-esi-acme-iptv-vlan-100)# tagged ethernet 1/10
```

Display ESI configuration

At this point, all ESIs have been set up. Enter the **show esi** command to display ESIs.

```
device(config)# show esi
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
acme            cvlan    2                                           0
foo             cvlan    2                                           0
santa           svlan    1                                           0
clara           svlan    1                                           0
```

In this display, ESIs are shown with their encapsulations and number of members (VLANs) in each ESI. At this stage, there are no client-provider bindings.

Associate customer ESI with provider ESI

Now that ESIs are defined for both the customer and provider sides, you can bind the customer ESI to the provider ESI using the **esi-client** command for the ESI named acme.

To complete the configuration for a IEEE 802.1ad network as shown in Sample configuration on page 234, associate the customer ESI to the provider ESI.

```
device (config)# esi santa
device (config-esi-santa)# esi-client acme
device(config-esi-santa)# exit
```

Show ESI command for the final configuration

Enter the **show esi** command to display the final configuration.

```
device(config)# show esi
ESI Name        Encap    Number of  Provider   Provider   Provider   Client
                         Members    ESI        Encap      VLAN       ESIs
--------        -----    ---------  --------   -----      --------   ------
acme            cvlan    2          santa        svlan      100        0
foo             cvlan    2          clara        svlan      200        0
santa           svlan    1                                           1
clara           svlan    1                                           1
```

In this display, the ESI named acme is an ESI of the encapsulation type CVLAN, which has two members (VLANs). The acme ESI now has the santa ESI as a provider with an SVLAN encapsulation type. The santa provider ESI is configured with VLAN ID 100.

This means that both CVLANs in the acme ESI receive a second SVLAN encapsulation (with a tag-type value of SVLAN as globally configured) and a SVLAN ID of 100.

PB using untagged members

For the configuration shown in Sample configuration on page 234, you added a Customer ESI and a Provider ESI and bound them together to provide PB service. This generic configuration can be used when a particular port is shared between customers. However, if a port is completely owned by one customer, you can use the following simple configuration to provide PB service.

Sample configuration 2

1. In this case, customer port 1/1 maps the customer's 4K VLANs. The provider side port 1/2 uses SVLAN 100. All customer traffic *from* the provider side is appended with an SVLAN 100 tag, and for all traffic going *to* the provider side will have the SVLAN 100 tag removed.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# port-type provider-network
```

2. Configure customer ports as provider network ports instead of customer edge ports.

```
device(config)# interface ethernet 1/2
device (config-if-e1000-1/1)# port-type provider-network
```

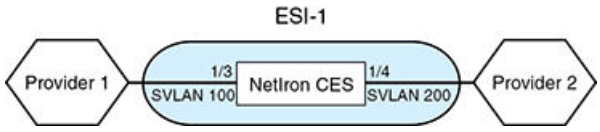
3. Add customer ports as untagged ports so that any traffic tagged with the 8100 tag will also be treated as untagged and will be appended with SVLAN 100.

```
device(config)# esi ESI_1
device(config)# esi ESI_1 encapsulation svlan
device(config-esi-ESI_1)# vlan 100
device(config-esi-ESI_1)# tagged ethernet 1/2
device(config-esi-ESI_1)# untagged ethernet 1/1
device(config-esi-ESI_1)# exit
```

SVLAN translation using flood domain configuration

The **single-flood-domain** command is used for SVLAN translation across provider domains.

FIGURE 32 SVLAN translation at an inter-provider boundary



In the configuration below, packets on port 1/3 with SVLAN=100 are translated to the port 1/4 with SVLAN=200 as the ports are in the same flood domain.

```
device(config)# esi ESI_1 encapsulation svlan
device(config-esi-ESI_1)# single-flood-domain
device(config-esi-ESI_1)# vlan 100
device(config-esi-ESI_1)# tagged ethernet 1/3
device(config-esi-ESI_1)# vlan 200
device(config-esi-ESI_1)# tagged ethernet 1/4
device(config-esi-ESI_1)# exit
```

Untagged ports

Packets from SVLAN port with only SVLAN tag (no CVLAN tag) are sent to only untagged ports in the default ESI.

Default VLAN

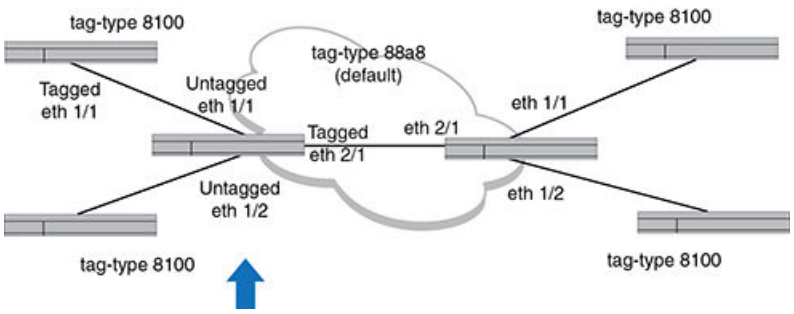
Packets from SVLAN with default VLAN as CVLAN tag are sent to untagged ports in the default ESI.

No Default VLANs are allowed inside non-default ESIs.

Port-based Service Interface Super AggregatedVLANs (SAV)

Using Port-based Service Interfaces is equivalent to using SAV in other Brocade products. Port-based Service Interfaces can be used when mapping a port based interface to a single Service VLAN Tag (S-TAG). When using a port-based service interface, it maps all VLAN-IDs from incoming ports to a SVLAN as shown in Port-based Service Interface Super AggregatedVLANs (SAV) on page 241.

Port-based service interface



Sample configuration

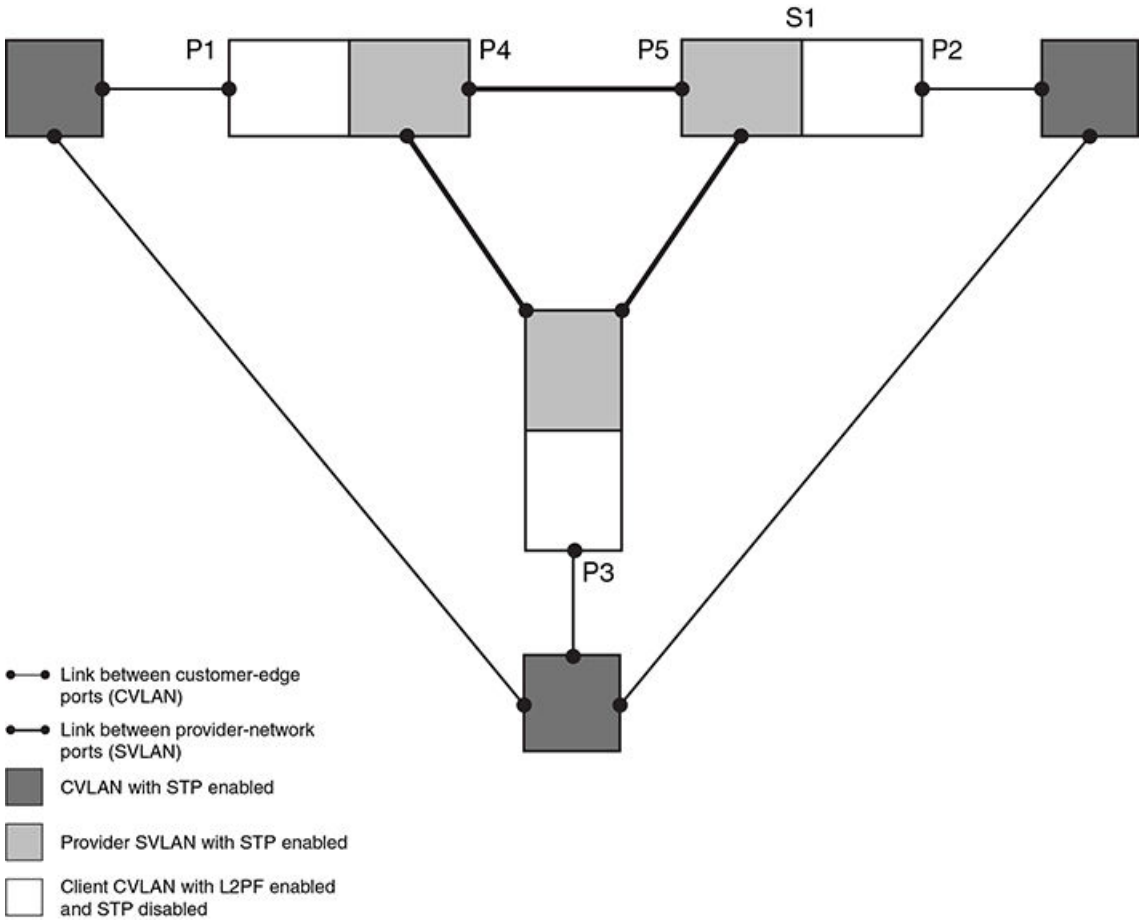
```
device(config)# tag-type tag2 ethernet 2/1
device(config)# vlan 100
device(config-vlan-100)# tagged ethernet 2/1
device(config-vlan-100)# untagged ethernet 1/1
device(config)# vlan 200
device(config-vlan-200)# tagged ethernet 2/1
device(config-vlan-200)#untagged ethernet 1/2
```

Layer 2 Protocol Forwarding (L2PF)

Layer 2 Protocol Forwarding (L2PF) is configured on CVLANs. You can configure the system to forward BPDUs on all CVLANs coming at different edge ports, drop all BPDUs, or selectively enable forwarding on a few CVLANs.

L2PF can transparently extend the STP topology of the CVLAN domain through the SVLAN domain, as if the CVLAN switches were directly hooked together. L2PF can be applied to CVLAN, which is a client of provider SVLAN as shown in "L2PF in a network".

FIGURE 33 L2PF in a network

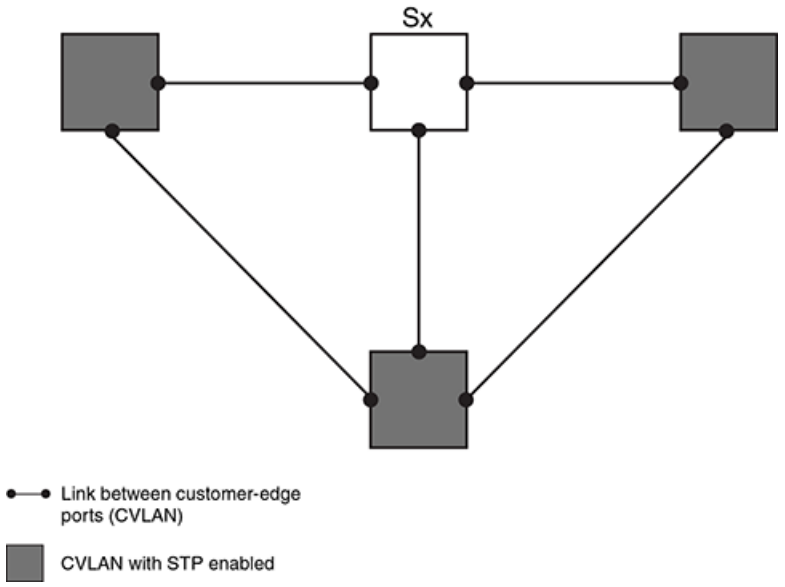


In this network, a CVLAN BPDU ingressing port P1 gets encapsulated and forwarded through the SVLAN domain as an ordinary multicast frame and then it egresses ports P2 and P3. In the SVLAN domain, a SVLAN BPDU originating from P4 and the ingressing port P5 gets terminated and processed in switch S1 for STP calculation. In other words, the final STP topology in the SVLAN domain is completely independent from the final STP topology in the CVLAN domain.

From the CVLAN domain's point of view, the physical network appears as shown in Figure 36.

Below, Sx is a simple bridge without STP.

FIGURE 34 Physical network



NOTE
It is critical for L2PF to disable STP on the client CVLAN to operate in that CVLAN.

STP wins over L2PF when both are enabled on a given client CVLAN. Table 44 displays the Port configuration for IEEE 802.1ah and IEEE 802.1ad.

TABLE 44 Port Configuration

Description	L2PF	STPF
CVLAN BPDU flooded inside CVLAN just like any multicast frame (but not flooded to SVLAN)	Disabled	Disabled
CVLAN BPDU terminated and processed in client CVLAN	Disabled	Enabled
CVLAN BPDU tunneled to provider SVLAN, and flooded in provider SVLAN	Enabled	Disabled
CVLAN BPDU terminated and processed in client CVLAN	Enabled	Enabled

NOTE
The behavior of L2PF is independent on whether the STP in the provider SVLAN is enabled or not.

Global configuration

By default, the device enables Layer 2 protocol forwarding (L2PF). Customer-BPDU packets on all CVLANs are forwarded to provider network ports when they arrive at customer-edge ports.

Since L2PF is globally enabled by default, all CVLANs have L2PF enabled by default. No CLI configuration is needed.

To globally disable Layer 2 protocol forwarding, enter the following command:

```
device(config)# no l2protocol-forwarding stp
```

To disable L2PF on a particular CVLAN, enter commands such as the following:

```
device(config)#esi acme encapsulation cvlan
device(config-abc)#vlan 10
device(config-abc-vlan-10)# no l2protocol-forward stp
```

Syntax: [no] l2protocol-forwarding stp

Use the **no** parameter to disable Layer 2 protocol forwarding.

IEEE 802.1ah Provider Backbone Bridging (PBB) Networks for the Brocade NetIron CES Series and the Brocade NetIron CER Series

Overview

The IEEE 802.1ah Provider Backbone Bridges (PBB) standard was developed to address the limitations of Provider Bridges (PB) and to add additional capabilities sought by Service Providers. When compared to a PB network, a PBB network deployment offers simplified operations, lower capital expenditures, and overall better scalability in terms of the number of supported customers. PBB also provides advantages when used in conjunction with VPLS, since PBB reduces the overall MAC address learning requirements. This section provides an overview of PBB, describes its advantages over PB, examines common PBB deployment scenarios, and examines its many benefits when deployed in combination with a core MPLS network supporting VPLS.

Provider Backbone Bridges

The Provider Backbone Bridges (PBB) standard, (IEEE 802.1ah), was developed to address the limitations of the Provider Bridges (PB) standard, (IEEE 802.1ad), and to add additional capabilities sought by Service Providers.

PB allows Service Providers to use a V-LAN identifier (VID) space separate from the customer VID (C-VID) space. PB adds a Service Provider VLAN Tag (S-TAG) containing a Service Provider VID (S-VID) to Ethernet frames (Figure 37). Because PB stacks a second VLAN tag to Ethernet frames, it is also known as "Q-in-Q," as a reference to the standard that originally defined VLAN tags, in other words, IEEE 802.1Q, which is known as defining "Q" frames.

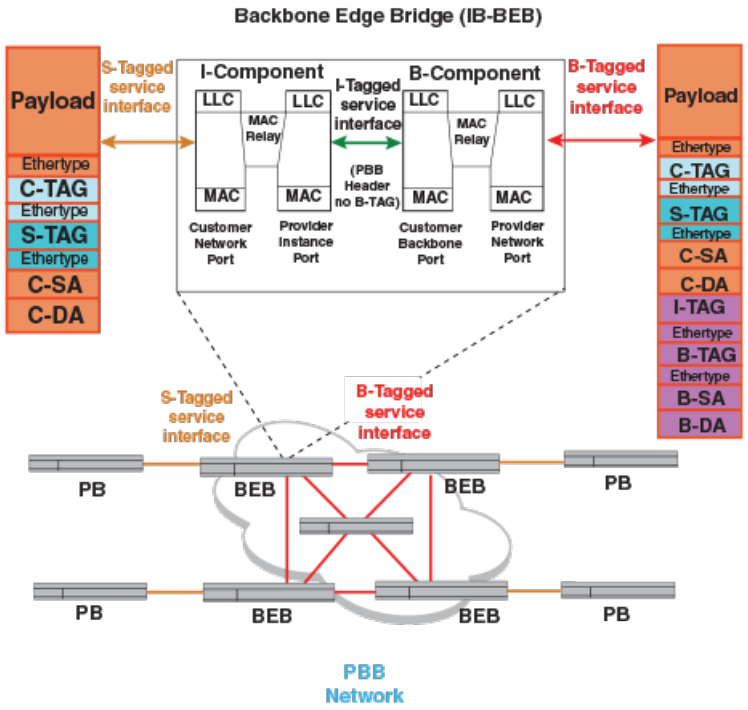
The S-VID field of the S-TAG is 12 bits long, which is the same length of a C-VID field of a customer VLAN Tag (C-TAG). Even though 12 bits can address up to 4096 distinct values, two values have special meaning and are reserved. Therefore, the Service Provider is limited to at most 4090 distinct S-VID values to identify service instances, that is, services or customers in a PB network. Another drawback is that PB frames are addressed by customer Media Access Control (MAC) addresses. This means that core Ethernet switches in a PB network have to learn all the source MAC addresses of all the customer frames traversing the core of the PB network. Thus, the size of the MAC address tables of core PB switches ultimately limits the number of customers that can be supported by a PB network.

To address the above described PB shortcomings, PBB adds a hierarchy view to Ethernet by encapsulating PB frames with a PBB header (which becomes the equivalent of a "Service Provider MAC header") containing a Backbone Destination MAC Address (B-DA), Backbone Source MAC Address (B-SA), and two new tags (Figure 37), which are described later in this document. What makes the B-DA and B-SA "backbone" addresses is the fact that these are MAC addresses of Service Provider's PBB edge switches. An edge PBB switch encapsulates an ingress PB frame with a PBB header containing the destination MAC address of an appropriate egress edge PBB switch. The egress edge PBB switch removes the PBB header and forwards the frame to an attached PB network. Because PBB adds a PBB header containing new destination and source MAC addresses, it is also known as "MAC-in-MAC."

By adding the PBB header, PBB isolates the Service Provider and customer address spaces. This means that Ethernet switches in the core of the Service Provider network will no longer learn customer MAC addresses or use customer MAC addresses to forward customer frames to their destinations. This improves the scaling of the Service Provider network in terms of the number of supported customers, since the number of supported customers is no longer directly tied to the size of the MAC address tables of the core Ethernet switches. In addition, the Service Provider network is now protected from customer network failures, since frame forwarding is now based on its own PBB header. Moreover, customers benefit from added security, since the customer's MAC addresses are no longer learned or used for frame forwarding decisions in the core of the Service Provider network.

As additional benefits to the Service Provider, PBB has the potential to simplify operations, e.g., by separating the customer and Service Provider addressing spaces, and to lower capital expenditures by reducing the cost of Ethernet switches used in the core of the network, since memory and processing power requirements are reduced by limiting MAC address learning to backbone MAC addresses.

FIGURE 35 Customer, PB, and PBB frame formats

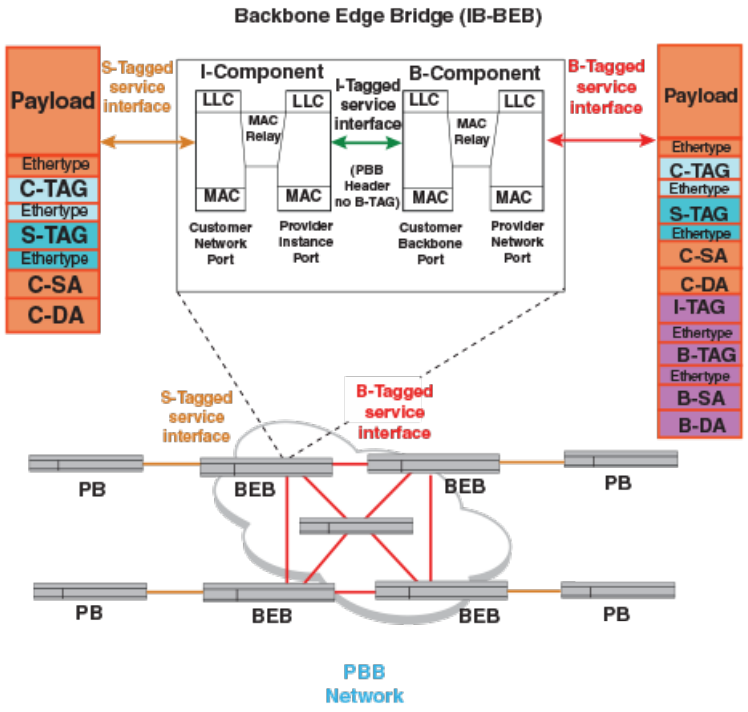


The new Backbone Service Instance Tag (I-TAG) contains a Backbone Service Instance Identifier (I-SID), which is 24 bits long. The I-SID field allows a Service Provider to identify up to 2 to the power of 24, that is, over 16 million service instances. In other words, over 16 million services or customers can be uniquely identified using the I-SID field. Therefore, PBB's I-TAG allows for highly scalable services by eliminating the 4090 service instances limitation of PB.

The semantics and the structure of the Backbone VLAN Tag (B-TAG) are identical to that of the PB S-TAG. The B-TAG was designed this way so that core PBB switches do not need to be aware of PBB. In fact, standard PB switches can be used in the core of a PBB network. Only the switches at the edge of the Service Provider PBB network need to be aware PBB.

A PBB network uses two types of bridges (Figure 38): Backbone Edge Bridges (BEB) and Backbone Core Bridges (BCB). As explained above, the functionality required from a BCB is the same as a standard IEEE 802.1ad PB bridge. A BEB is used at the boundary of a PBB network to add and remove the PBB header.

FIGURE 36 Backbone Edge Bridge operation



As defined in IEEE 802.1ah, a BEB has two main components: an I-Component and a B-Component. From left to right in Figure 38, the I-Component maps S-VIDs to I-SIDs and adds a PBB header without a B-TAG, while the B-Component maps I-SIDs to B-VIDs and adds a B-TAG. These actions are reverted in the opposite direction.

As shown in Figure 38, a BEB containing an I-Component and a B-Component is called an IB-BEB. The B-Component of an IB-BEB forwards frames towards the core of a PBB network based on backbone MAC addresses (that is, it learns backbone MAC addresses), while the I-Component forwards frames towards the PB network based on customer MAC addresses (that is, it learns customer MAC addresses). The terms I-BEB and B-BEB refer to optional BEBs that support a single component type, that is, either I-Component or B-Component, respectively. I-BEBs and B-BEBs expose an I-Tagged service interface, which carries frames with a PBB header, but without a B-TAG.

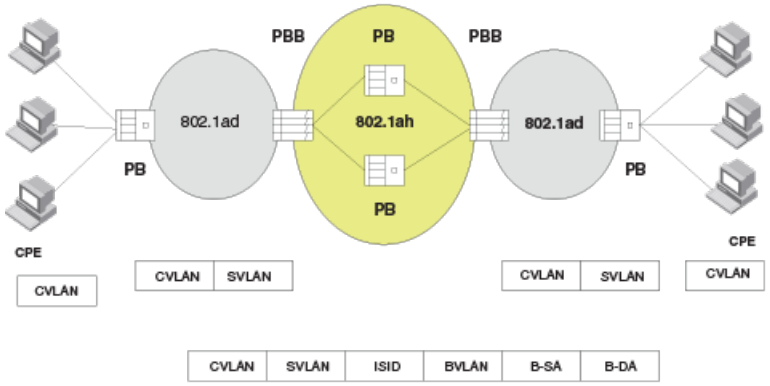
As with PB, PBB networks use a Spanning Tree Protocol (STP), e.g., RSTP or MSTP, to dynamically determine the active topology of the PBB network and MAC address learning to dynamically build a forwarding database. Since an IB-BEB forwards frames to PB and PBB networks, it has to learn customer and backbone MAC addresses. However, since an IB-BEB is at the edge of the Service Provider network, it only learns customer MAC addresses of the local traffic.

NOTE
When using ESI VLANs in the configuration, always configure the protocol in default VLAN.

IEEE 802.1ah Provider Backbone Bridging (PBB)

This section provides details on IEEE 802.1ah Provider Backbone Bridging (PBB), with a description and a configuration example of an integrated PB and PBB network.

FIGURE 37 Integrated IEEE 802.1ad and IEEE 802.1ah network architecture



The Provider Backbone Bridge (PBB) protocol typically resides at the core of a carrier network, interconnecting PB networks. Inside of a PBB network a carrier usually deploys PB systems for layer 2 interconnectivity. The PBB protocol is designed to insulate carrier infrastructure from having to learn customer MAC addresses and provide a separation of service and networking components of an Ethernet service provided to the customers:

- It provides IEEE 802.1ah encapsulation to add a backbone MAC header on the incoming PB packet (which has customer DA or SA) so core carrier switches don't need to learn customer MAC addresses.
- It supports creation of EVC (Ethernet Virtual Circuits) with a concept of an Ethernet Service Instance (ESI), which is enabled by use of a globally unique 24-bit I-component Service Identifier (ISID).
- PB packets are encapsulated with ISID, BVLAN, B-SA, and B-DA as shown in Figure 39.
- Layer 2 switches in the IEEE 802.1ah-encapsulated network are normal PB systems, which process BVLAN header as if it was an SVLAN-encapsulated packet.
- **Tag-types** : BVLAN and SVLAN tag-types are same (default 0x88a8), so the PB systems inside a PBB network can process BVLAN packets as if they were SVLAN encapsulated.
- **ISID operation** : A 24-bit ISID supports the provider bridging operation framework where an EVC setup gives the closed environment for customer packets on the two ends to be limited to the EVC. The EVC is identified by the ISID value.
- Customer MAC address learning is limited to the ISID domains that are part of an EVC. A PBB device doesn't need to learn customer MAC addresses that are not part of this EVC. This provides scalability, as devices are not required to store every customer MAC address that passes through the provider backbone network.
- **BVLAN** : BVLAN provides the network connectivity among PBB nodes. Notice that while in a PB network the SVLAN provides both the networking operation and service interconnection, in a PBB network the service provisioning (ISID) is totally independent on networking framework (BVLAN).

IEEE 802.1ah configuration options

Two types of architectures are possible for providing connectivity to PBB networks:

- Ingress ports receive SVLAN-encapsulated packets. In this case, the ingress port already contains SVLAN mapping and the PBB system provides additional encapsulation for the PBB header.
- Ingress ports directly connect to customer devices and receive CVLAN-mapped traffic. The system then performs SVLAN mapping internally and then performs IEEE 802.1ah encapsulation before sending packets out to PBB network. This model is not supported in Brocade NetIron CES Series and Brocade NetIron CER Series.

Tag type configuration

Tag Type values for different encapsulations are defined globally for all types of VLANs.

At most, two different external tag-types can be defined for Brocade NetIron CES Series. The restriction of two tag types doesn't include an additional tag-type that can be defined for ISID. ISID encapsulation is performed internally and doesn't have a port association.

IEEE-defined tag-type values are as follows:

- CVLAN: 8100
- SVLAN & BVLAN: 88a8 (both PBB and PB bridges have identical outer tag-type so core PB bridges process BVLAN tags and interoperate with PBB bridges)
- ISID: 88e7

When configuring tag types, all tag types must be entered in **a single command line** , as shown below.

```
device(config)# tag-type cvlan tag1 svlan tag2 bvlan tag2
```

Displaying tag types

To display the tag types, enter the following command.

```
device(config)# show tag-type
Encap      Current VLAN Tags             Default VLAN Tags
-----      -----------------             -----------------
cvlan                 8100                     8100
svlan                 9100                     88A8
isid                  86B5                     88E7
bvlan                 9100                     88A8
```

NOTE
On PB networks, SVLAN and BVLAN tag types must be same.

Port configuration for IEEE 802.1ah and IEEE802.1ad at each interface

Table 45 displays the Port configuration for IEEE 802.1ah and IEEE 802.1ad.

TABLE 45 Port configuration for IEEE 802.1ah and IEEE 802.1ad

Port type	Description	Characteristics
PB_CE	Customer Edge Port (IEEE 802.1ad). This is the default port type.	• CVLAN tag.
PB_PN	Provider Network Port (IEEE 802.1ad)	• SVLAN tag • IEEE 802.1ad frame at this interface.
PBB_BE	Backbone-Edge (IEEE 802.1ah)	• SVLAN tag • IEEE 802.1ad frame • This is same encapsulation type as PN, but the provider side of the port (into the carrier network) is an IEEE 802.1ah frame. • A BE port connects to a PB network.
PBB-BN	Backbone-Network (IEEE 802.1ah)	• BVLAN tag

IEEE 802.1ah Provider Backbone Bridging (PBB)network configuration example

The Brocade NetIron CES Series and Brocade NetIron CER Series sample configuration for PBB functionality is shown in Figure 40. The Brocade NetIron CES Series and Brocade NetIron CER Series take in SVLAN inputs, map internally to an ISID, and then bind to a BVLAN to provide PBB functionality.

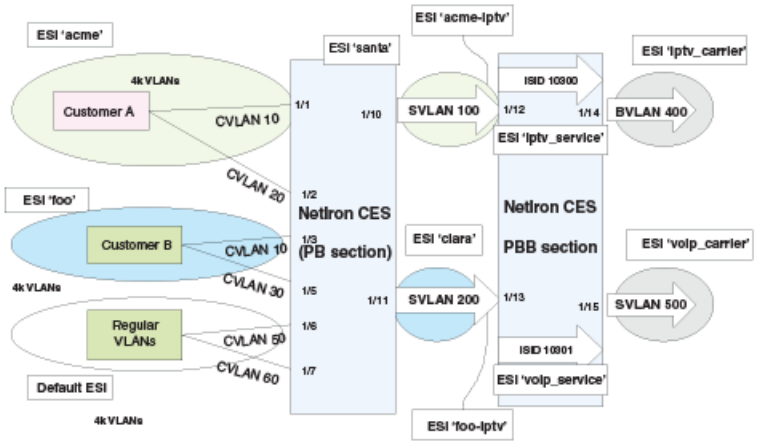
In Figure 40, PB output with SVLAN encapsulation using ESI 'santa' is used to provide input to Ethernet port 1/12 which is configured as a backbone-edge port.

A new ESI 'acme-iptv' is created for the incoming SVLAN (at VLAN ID = 100). This is assigned to a BVLAN (VLAN ID = 400) under ESI 'iptv_carrier' by first mapping it to ISID 10300 under ESI 'iptv_service'.

In this example, PB output with SVLAN encapsulation using ESI 'santa' is used to provide input to Ethernet port 1/12 which is configured as a backbone-edge port.

A new ESI 'acme-iptv' is created for the incoming SVLAN (at VLAN ID = 100). This is assigned to a BVLAN (VLAN ID = 400) under ESI 'iptv_carrier' by first mapping it to ISID 10300 under ESI 'iptv_service'.

FIGURE 38 Provider Backbone Bridging (PBB) functionality



IEEE 802.1ah configurations

The PBB protocol performs IEEE 802.1ah encapsulation on packets that arrive from PB network and are SVLAN-tagged so core carrier switches don't need to learn customer MAC addresses.

Two types of architectures are possible for providing connectivity to PBB networks.

Sequential steps for a IEEE 802.1ah configuration CLI example are shown below.

Define interface types

First step in setting up a PB network configuration is to set interface type properly. Interface type has to match the encapsulation type of the VLAN expected on the interface.

By default, interfaces are of type 'customer-edge' so there is no need to define an interface type for CVLAN ESIs.

1. Set 1/12 and 1/13 to 'backbone-edge' (SVLAN).

```
device(config)# interface ethernet 1/12
device(config-if-e10000-1/10)# port-type backbone-edge
device(config-if-e10000-1/10)# exit
```

2. Configure port 1/14 to be of 'backbone-network' type.

```
device(config)# interface ethernet 1/14
device(config-if-e10000-1/10)# port-type backbone-network
device(config-if-e10000-1/10)# exit
```

Create an SVLAN ESI on IEEE 802.1ad side (PBB ingress)

1. Create an ESI on 801.1ad side. Acme-iptv" is name of the provider IEEE 802.1ad service.

```
device(config)# esi acme-iptv encapsulation svlan
```

2. Define SVLAN 100 as one of the parameters for ESI Acme-iptv.

```
device(config-esi-acme-iptv)# vlan 100
```

3. Associate physical ports to SVLAN 100.

```
device(config-esi-acme-iptv-vlan-100)# tagged ethernet 1/12
device(config-esi-acme-iptv-vlan-100)# exit
```

Create PBB ESI for ISID (PBB ingress - BEB function)

1. Create an ESI on PBB for ISID. Configure "iptv-carrier" as the name of the carrier service.

```
device(config)# esi iptv-service encapsulation isid
```

2. Define any special parameters for this ESI.

```
device(config-esi-iptv-service)# isid 10300
device(config-esi-iptv-service-isid-10300)# exit
```

Create PBB ESI on the carrier side (BVLAN)

1. Create an ESI on PBB. Configure "iptv-carrier" as the name of the carrier service providing IEEE 802.1ah.

```
device(config)# esi iptv-carrier encapsulation bvlan
```

2. Define any special parameters for this ESI.

```
device(config-esi-iptv-carrier)# vlan 400
```

3. Port configuration.

```
device(config-esi-iptv-carrier-vlan-400)# tagged ethernet 1/14
```

Bind ISID to BVLAN

Specify that ISID ESI "iptv-service" is a client of BVLAN ESI "iptv-carrier" and binding is done in two steps. The first command below binds SVLAN ESI 'santa' to ISID 'iptv-service' then puts the ISID inside BVLAN. The second command binds SVLAN ESI 'clara' to ISID 'iptv-service' then puts the ISID inside BVLAN.

1. Bind SVLAN ESI 'santa' to ISID 'iptv-service' then puts the ISID inside BVLAN.

```
device(config-esi-iptv-service)# esi-client acme-iptv isid iptv-service
```

2. Bind SVLAN ESI 'clara' to ISID 'iptv-service' then puts the ISID inside BVLAN.

```
device(config-esi-iptv-service)# esi-client clara isid voip-service
```

ESI configuration display after mappings

To display the ESI configurations, enter the **show esi** command.

```
device(config)#show esi
ESI Name      Encap    Number of  Provider   Provider   Provider   Client
                       Members    ESI        Encap      VLAN       ESIs
--------      -----    ---------  --------   -----      ---------  ------
acme          cvlan    2          santa        svlan      100      0
foo           cvlan    2          clara        svlan      200      0
santa         svlan    1          iptv-service isid       10300    1
clara         svlan    1          iptv-service isid       10300    1
iptv-service  isid     1          iptv-carrier bvlan      400      2
iptv-carrier  bvlan    1          None         None       None     1
```

Syntax: show esi

```
device(config-esi-foo)#
ESI: acme Encapsulation: cvlan
PORT-VLAN 42, Name [None], Priority Level0
L2 protocols   : NONE
ESI: acme Encapsulation: cvlan
PORT-VLAN 43, Name [None], Priority Level0
L2 protocols   : NONE
ESI: acme Encapsulation: cvlan
PORT-VLAN 10, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/3
ESI: foo Encapsulation: cvlan
PORT-VLAN 30, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/5
ESI: foo Encapsulation: cvlanv
PORT-VLAN 100, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/10
ESI: santa Encapsulation: svlan
PORT-VLAN 200, Name [None], Priority Level0
L2 protocols   : NONE
Tagged Ports   : ethe 1/11
```

Integrated IEEE 802.1ad and IEEE 802.1ah

The Provider Backbone Bridge (PBB) protocol usually resides at the core of the network, interconnecting Provider Bridging networks. Integrated IEEE 802.1ad and IEEE 802.1ah provides the following:

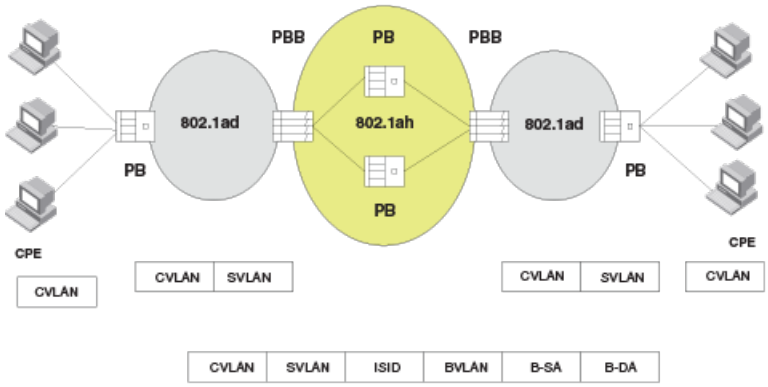
- Provides IEEE 802.1ah encapsulation to add a backbone MAC header on the incoming Provider Bridging packet so the carrier switches will not need to learn the customer MAC address.
- Supports the creation of Ethernet Virtual Circuits (EVC) with a concept of an Ethernet Service Instance (ESI), which is enabled by use of a globally unique 24-bit 1-component Service Identifier (ISID).
- Layer 2 switches in the IEEE 802.1ah-encapsulated network are normal Provider Bridging systems, which process BVLAN header as if it was an SVLAN-encapsulated packet.

- **Tag-types** - BVLAN and SVLAN tag-types are the same (default 0x88a8), so the Provider Bridging system inside the PBB network can process BVLAN packets as they were SVLAN encapsulated.
- ISID Operation:
  - 24-bit ISID supports the provider bridging operation framework where an Ethernet Virtual Circuits (EVC) setup gives the closed environment for customer packets on the two ends to be limited to the EVC.
  - Customer MAC address learning is limited to the ISID domains that are part of an EVC. A PBB device does not need to learn customer MAC addresses that are not part of this EVC. This provides scalability, as devices are not required to store every customer MAC address that passes through the provider backbone network.
- **BVLAN:** BVLAN provides the network connectivity among PBB nodes.

NOTE
While in a Provider Bridging network the SVLAN provides both the networking operation and service interconnection, in a PBB network the service provisioning (ISID) is totally independent on networking framework (BVLAN).

Figure 41 provides an example of a Provider Backbone Bridged (PBB) network interconnecting two Provider Bridged (PB) networks.

FIGURE 39 Provider Backbone Bridged network interconnecting two Provider Bridged networks

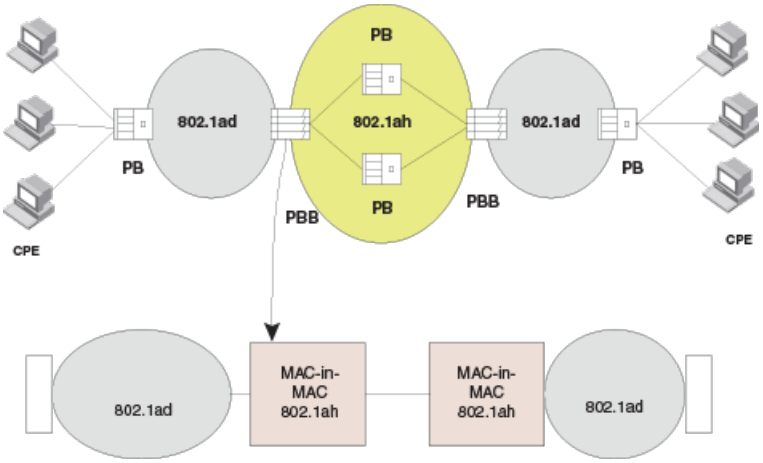


IEEE 802.1ah (PBB) configurations

Ingress ports receive SVLAN-encapsulated packets. In this case, the ingress port already contains SVLAN tagged frames and the PBB switch provides additional encapsulation by adding the PBB header. An SVLAN can be mapped to one unique ISID or alternatively multiple SVLANs can be mapped to the same ISID.

In Figure 42, the PBB device expects tagged Ethernet packets coming in with SVLAN encapsulation. This is the most common configuration for PBB as a provider of carriers' backbone infrastructure.

FIGURE 40 IEEE 802.1ah PBB configuration



Interface configuration for Provider Bridge and Provider Backbone Bridge (PBB) networks

When configuring a Brocade device, a port is configured to be one of the one of the interface types:

- – Customer-edge (CE)
  – Provider-network (PN)
  – Backbone-edge (BE)
  – Backbone-network (BN)

Configuring the port-type for an interface

Before a VLAN can be provisioned for an interface, the port-type for the interface must be defined. This command defines a port type for an Ethernet interface. The port-types specify both sides of IEEE 802.1ad and IEEE 802.1ah networks. To define the port-type of the interface, enter commands such as the following.

```
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# port-type provider-network
device(config-if-e10000-5/1)#
```

Syntax: **port-type** [**backbone-edge** | **backbone-network** | **customer-edge** | **provider-network**]

Use the **backbone-edge** parameter to specify the Backbone Edge Port for IEEE 802.1ah PBB

Use the **backbone-network** parameter to specify the Backbone Network Port for IEEE 802.1ah PBB

Use the **customer-edge** parameter to specify the Customer Edge Port for IEEE 802.1ad PB

Use the **provider-network** parameter to specify the Provider Network Port IEEE 802.1ad PB

Displaying port- types

The **show interfaces** command displays port-type for an interface, as shown below.

```
device(config-if-e10000-5/1)# show interfaces
10GigabitEthernet5/1 is empty, line protocol is down
  Hardware is 10GigabitEthernet, address is 0e04.80de.ada0 (bia 0e04.80de.ada0)
  Configured speed 10Gbit, actual unknown, configured duplex fdx, actual unknown
  Member of VLAN 1 (untagged), port is in untagged mode, port state is Disabled
```

```
STP configured to ON, Priority is level0, flow control enabled
arp-inspection-trust configured to OFF
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
Port-type (802.1ad/802.1ah): provider-network
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
NP received 0 packets, Sent to TM 0 packets
NP Ingress dropped 0 packets
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions
NP transmitted 0 packets, Received from TM 0 packets
```

Syntax: show interfaces

TABLE 46 Display of show interfaces command

This field...	Displays...
Module type Port# is *State*	The *module type* variable specifies a type of interface module, such as 10GigabitEthernet. The *port#* variable specifies the port number for the interface module. The *state* variable if the interface module is up or down.
Line protocol is *status*	The *status* variable specifies if the line protocol is up or down. If the interface is down due to Link Fault Signaling - Remote Fault Notification (LFS or RFN) the reason is indicated as: "(remote fault)".
STP Root Guard is *status*	The *status* variable specifies if the STP Root Guard is enabled or disabled.
STP BPDU Guard is *status*	The *status* variable specifies if the STP BPDU Guard is enabled or disabled.
Hardware is *module type*	The *module type* variable specifies a type of interface module, such as # Gigabit Ethernet.
Address is *MAC- address*	The *MAC- address* variable specifies the MAC address of the port.
Configured speed and actual speed	The speed that the module has been configured to operate at, and the actual speed at which it is operating.
Configured port speed and actual duplex	The port capacity that the module has been configured to operate at, and the actual speed at which it is operating.
Member of *VLAN #* (untagged) *port#* L2 VLANS (tagged) Port is in dual mode/untagged/tagged mode Port state is *status*	The *VLAN#* (untagged) variable specifies a port that is a member of only 1 VLAN. The *port#* Layer 2 VLANS (tagged) variable specifies a port that is a member of multiple ports and untagged. A port is in *dual- mode* specifies member VLAN ports as untagged and tagged. The default mode is dual-mode. The *status* variable identifies the flow of traffic as forwarding or disabled.
STP configured to *status* Priority level Flow control *status*	The *status* variable specifies if the STP is ON or OFF. The priority level assigned to the port-based VLAN. The priority level is on scale from 0-7. The default is 0. The *status* variable is enabled or disabled.
Priority force *status*	The *status* variable specifies if the priority force on a port is disabled on enabled.
Drop precedence level *value*	Identifies the TOS or DSCP value in the IPv4 or IPv6 packet header.

TABLE 46 Display of show interfaces command (continued)

This field...	Displays...
	The *value* variable specifies the drop precedence on a scale from 0-3. Packets that contain a DSCP value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped. The default value is 0.
Drop precedence force *status*	The *status* variable specifies the drop precedence force as enabled or disabled. Identifies the drop precedence if the force command is configured for a specific ingress port.
arp-inspection-trust configured to *status*	The *status* variable specifies if arp-inspection-trust feature is configured ON or OFF. The default trust setting for a port is untrusted.
Mirror *status*	The *status* variable specifies if the port mirror command is configured as enabled or disabled.
Monitor *status*	The *status* variable specifies if the port monitor command is configured as enabled or disabled.
Trunk membership	The *Trunk membership* variable identifies the interface module as a member of a primary or secondary port. This specifies members of an active port or not a member of an active port.
Configured trunk membership	The *Configured trunk membership* variable identifies the interface module as a member of any configured trunk or not a member of a configured trunk.
Port name	The *port name* variable identifies the name assigned to the port.
MTU *# bytes* , encapsulation ethernet	Maximum Transmission Unit (MTU) refers to the size of the largest packet or frame that a known layer can pass forward. The *# bytes* variable refers to size of the packet or frame.
#seconds input rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* input rate refers to: • The *value* of bits received per second. • The *value* of packets received per second. • The *%* utilization specifies the port's bandwidth used by received traffic.
# seconds output rate: *value* bits/sec, *value* packets/sec, *%* utilization	The *#second* output rate refers to: • The *value* of bits transmitted per second. • The *value* of packets transmitted per second. • The *%* utilization specifies the port's bandwidth used by transmitted traffic.
value packets input, *value* bytes	• The *value* variable specifies the number of packets received. • The *value* variable specifies the number of bytes received.
Received *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module is receiving on broadcasts, multicasts, and unicast traffic.
value input errors, *value* CRC, *value* frame, *value* ignored	• The *value* variable specifies the number of received packets with errors. • The *value* variable specifies the number of received packets with CRC errors. • The *value* variable specifies the number of received packets with alignment errors. • The *value* variable specifies the number of received packets that are discarded.
value runts, *value* giants	The *value* runts variable specifies the number of small packets that are less than 64 bytes.

TABLE 46 Display of show interfaces command (continued)

This field...	Displays...
	The *value* giants variable specifies the number of large packets greater than 1518 bytes.
NP received	The number of packets received on the Network Processor (NP).
NP transmitted	The number of packets sent from the Network Processor to the Traffic Manager (TM).
NP Ingress dropped	The number of ingress packets dropped on the Network Processor.
value packets output *value* bytes	• The *value* variable specifies the number of transmitted packets. • The *value* variable specifies the number of transmitted bytes.
Transmitted *value* broadcasts, *value* multicasts, *value* unicasts	The *value* variable specifies the amount of traffic the interface module transmitted on broadcasts, multicasts, and unicast traffic.
value output errors, *value* collisions	• The *value* variable specifies the number of transmitted packets with errors. • The *value* variable specifies the number of transmitted packets with collision errors.
Network Processor transmitted *value* packets Received from Traffic Manager *value* packets	The *value* variable specifies the number of packets transmitted from the Network Processor. The *value* variable specifies the number of packets received by the Network Processor from the Traffic Manager.

Point to Point PBB

Point to Point PBB (P2P PBB) provides the ability to turn off MAC address learning on a per service instance basis (ISID). This provides support for point-to-point services, such as EVPL, which do not require MAC address learning. Point to Point PBB is designed to flood the traffic to a specific remote BEB in the core PBB network, instead of flooding across all the BEBs.

Limitations

When P2P PBB is enabled, the MACs learned on S-VLANs are duplicated in the new flood domain. This will reduce the maximum number of supported MACs from 131072 VPLS MACs to 65536 VPLS MACs.

Configuring Point to Point PBB

Use the **p2p-mac** command to enable this feature. Since the P2P PBB feature is specific to a service instance it has to be executed for each ISID.

```
device(config)#esi A-isid encapsulation isid
device(config-esi-A-isid)#isid 18001
device(config-esi-A-isid-isid-18001)#p2p-mac 001b.edb4.5ac1
```

Syntax: [no] p2p-mac *mac-addr*

The *mac-addr* parameter requires the MAC address of the remote BEB.

Use the **no** command to disable this feature.

Show commands

Use the **show flood-domain** command to display the extra flood-domain (shadow flood domain). This extra flood domain information is only shown for P2P PBB instances.

```
device# show flood-domain
FDID                Type             NumMem             VLAN Owner      VLAN Owner ESI
----                ----             ---------      ----------      ---------
4357                PBB               2                    1234               i-isid
4358                B_VLAN        1                        500               b-vlan
4359                S_LOOP        1                        100               s-vlan
4360                S_FWD         2                       100               s-vlan
4369, 4370      PBB                          1               2345               A-isid
```

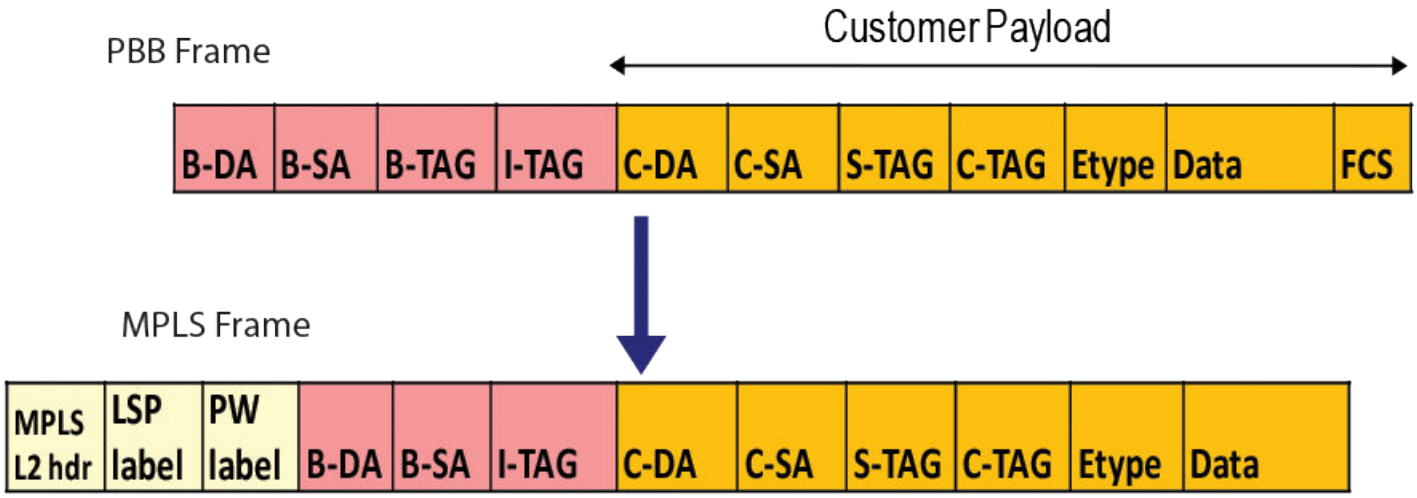
You can use the **show flood-domain** command with the *fd_id* variable to filter the flood domain. The **show flood-domain** command with the *shadow_fd_id* variable can also be used to show the identical information.

```
device#show flood-domain 4369
device#show flood-domain 4370
FDID                Type           NumMem               VLAN Owner      VLAN Owner ESI
----                ----          ----------     ----------       ---------
4369, 4370      PBB                     1               2345                A-isid
```

ISID mapping to VPLS

The ISID mapping to VPLS feature allows the service instance to be identified end-to-end across the Ethernet and VPLS networks using the same value without modifying how the MPLS network operates. When the PBB packet is sent out on the MPLS cloud, the ISID is always preserved in the packet as the payload tag. A VPLS instance can recognize PBB packets only when it is configured in tagged-mode. Figure 43 illustrates this feature.

FIGURE 41 ISID mapping

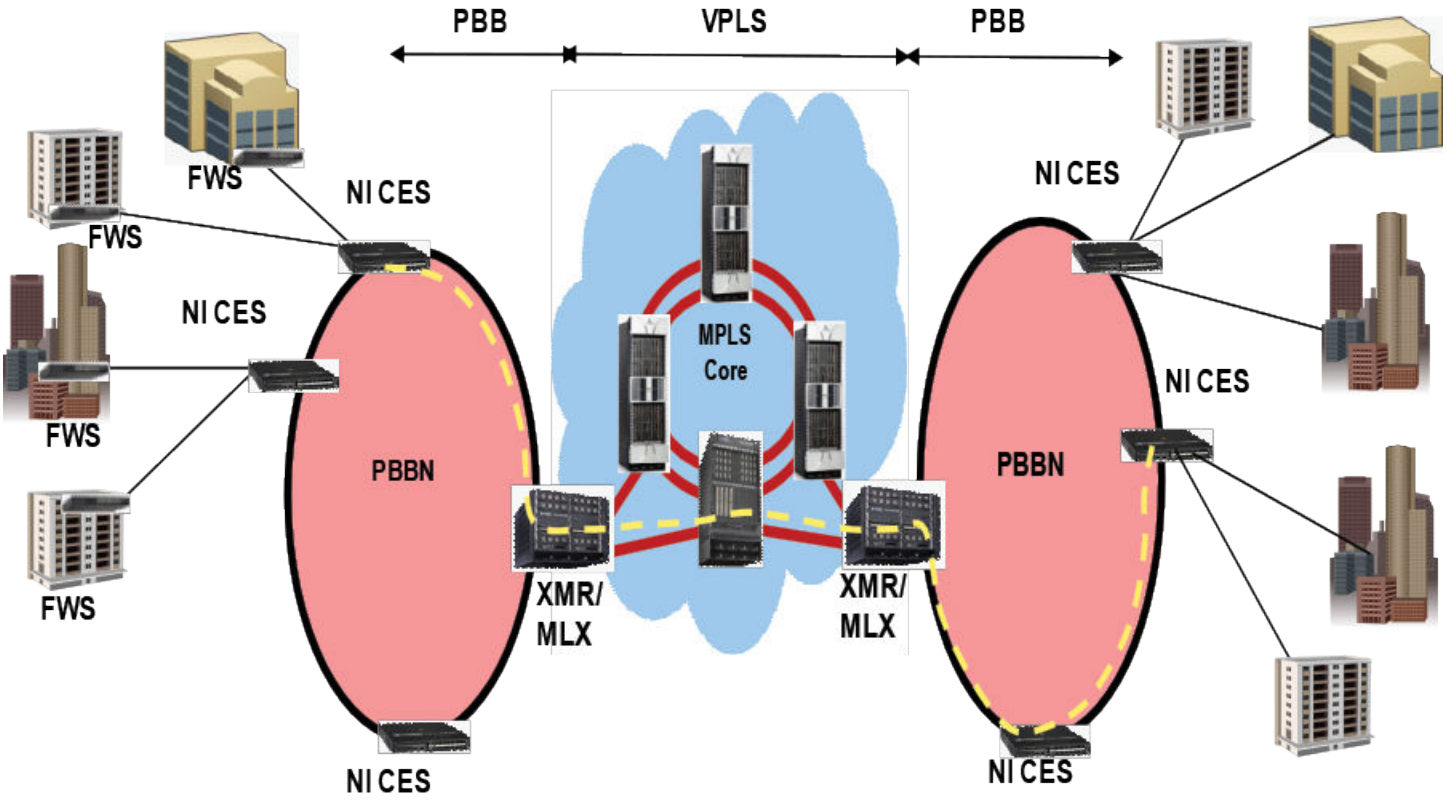


ISID endpoint configuration considerations

- ISID range is between 256 (0x100) and 16777214 (0xFFFFFE).
- ISID endpoints can be configured only if the VPLS instance is using the tagged mode.

- You cannot mix ISID with non-ISID endpoints in the same VPLS instance.
- All endpoints within a VPLS instance should have the same ISID.
- For remote switching, you must configure the same ISID on both the ingress and egress routers.
- You cannot configure more than one ISID endpoint on the same port in the same VPLS instance.
- Multicast snooping will not be allowed on a VPLS instance which contains ISID endpoints.
- The **default-max-frame-size** configured must be sufficient to carry the entire packet across the MPLS cloud.

FIGURE 42 ISID endpoint configuration example



Configuring the ISID endpoints

The existing VLAN CLI under the VPLS configuration mode has been enhanced to configure ISID endpoints.

To configure ISID endpoints, enter commands such as the following:

```
device(config-mpls)# vpls test 100
device(config-mpls-vpls-test)# vc-mode tagged
device(config-mpls-vpls-test)# vlan 100 isid 450
```

Syntax: [no] vlan 1-4094 [inner-vlan 1-4095 | isid 256 - 16777214]

The outer vlan can be from 1 to 4094, but it excludes the default VLAN ID.

The **inner-vlan** can be in the range from 1 to 4095.

The **ISID** can be in the range from 256 to 16777214. The ISIDs from 0 to 255 and 16777215 are reserve.

Tag type and ether type

For a packet to be classified as a Dual Tag packet, the Ether type of the inner-vlan tag in the packet should always be 0x8100. The expected Ether type for the outer vlan (B-TAG/S-TAG) can be changed using the global **tag-type** command. The default is 0x8100. Different Ether types can be configured per-port.

The existing tag-type CLI has been enhanced to specify the expected Etype for I-TAG. The default is 0x88e7 (this Etype configuration is global and cannot be specified per-port). The configuration will take effect immediately, and you do not need to reload the box for the new Ether type to take effect.

```
device(config)# tag-type)
device(config)# tag-type isid 88e8
```

Syntax: [no] tag-type *HEX-VALUE* [*port-type slot/port* [**to** *slot/port*]

Use the *HEX-VALUE* parameter to specify the etype in hex (Default: 0x8100).

Syntax: [no] tag-type isid *HEX-VALUE*

Use the **isid** parameter to specify the etype for I-Tagged packets.

Use the *HEX-VALUE* parameter to specify the etype in hex (Default: 0x88e7).

Topology Groups

For topology groups, the member VPLS VLAN commands have been enhanced to specify the ISID level. To configure member VPLS VLAN at the ISID level, enter commands such as the following.

```
device(config)# topo 1
device(config-topo-group-1)# master-vlan 100
device(config-topo-group-1)# member-vlan vpls id 100 vlan 100 isid 450
```

Syntax: [no] member-vlan vpls id *vpls-id* | **name** *name* **vlan** *vlan-id* [**to** *vlan-id*] **isid** *isid*

OR

Syntax: [no] member-vlan vpls id *vpls-id* | **name** *name* **vlan** *vlan-id* [**inner-vlan** *inner-vlan-id* [**to** *inner-vlan-id*] ... | **isid** *isid*]

Show commands

Use the **show mpls vplsb id** command to display ISID information.

```
device# show mpls vpls id 3
VPLS name_raw, Id 3, Max macentries: 8192
Total vlans: 1, Tagged ports: 3 (3 Up), Untagged ports 0 (0 Up)
IFL-ID: 4097
Vlan300 inner-vlan500
Tagged: ethe3/1 ethe3/11 ethe3/13
Total VC labels allocated: 16 (983072-983087)
VC-Mode: Raw
Total VPLS peers: 1 (1 Operational)
Peer address: 200.200.200.200, State: Operational, Uptime: 1 hr 10 min
Tnnlin use: tnl1(4)
LDP session: Up, Local VC lbl: 983072, Remote VC lbl: 983072
Local VC MTU: 1500, Remote VC MTU: 1500
LOCAL VC-Type: Ethernet (0x05), Remote VC-Type: Ethernet (0x05)
CPU-Protection: OFF
Local Switching: Enabled
```

To display the ISID endpoints in a topology group, use the following command.

```
device# show topology
Topology Group 1
==================
Topo HW Index   : 65535
Master VLAN     : 100
VPLS VLAN exist : TRUE
Member VLAN     : None
Member Group  : None
M
ember VPLSs  : vpls id 100 vlan 100 isid 450

Control Ports : None
Free Ports :
VPLS ID 100 VLAN 100 ISID 450 - ethe 2/1
```

Syntax: show topology

Load balancing traffic

You can enable the device to mask out different PBB header fields. When a PBB packet is received on the device, by default the network processor checks the packet (including the Bvlan SA/DA) and includes the PBB header fields in hash calculations even when an ISID endpoint is not configured on the device.

To mask out the PBB header fields during hash calculations, use the **load-balance** command.

```
device(config)# load-balance mask pbb cust-l2-hdr all
device(config)# load-balance mask pbb cust-ip4-ip6-hdr 1
device(config)# load-balance mask ethernet isid 2 1
```

Syntax: [no] load-balance mask ethernet *sa-mac* | *da-mac* | *vlan* | *etype* | *inner-vlan* | *isid* **all** | *slot* [*np-id*]

Syntax: [no] load-balance mask pbb *cust-l2-hdr* | *cust-ip4-ip6-hdr* **all** | *slot* [*np-id*]

By default all options are off.

When using the **isid** option for the **load-balance mask ethernet** command, the ISID field in the PBB packet on the endpoint will be masked out.

When using the **cust-l2-hdr** option for the **load-balance mask pbb** command, the destination and source MAC addresses in the customer Layer 2 header will be masked out during hash-calculations. This applies only to the ingress device, for packets received on the endpoint.

When using the **cust-ip4-ip6-hdr** option for the **load-balance mask pbb** command, the IPv4 and IPv6 protocol field and destination and source addresses in the customer Layer 3 header will be masked out. This applies only to the ingress device, for packets received on the endpoint.

> NOTE
> During the hash calculations, cust-l2-hdr and cust-ip4-ip6-hdr will not be considered.

When using the **all** option, it will turn on the masking on all cards and packet processors.

The **slot** option will turn on the specific slot, and when combined with the *np-id*, it will be enabled on the specific packet processor of a given slot.

Syntax: [no] load-balance mask ethernet *sa-mac* | *da-mac* | *vlan* | *etype* | *inner-vlan* | *isid* **all** | *slot* [*np-id*]

Syntax: [no] load-balance mask pbb *cust-l2-hdr* | *cust-ip4-ip6-hdr* **all** | *slot* [*np-id*]

Show commands

Use the **show load-balance mask-option** command to view the mask sub-options individually.

```
device# show load-balance mask-options ethernet
Mask Ethernet options -
Mask Source MAC is enabled on -
No Slots
Mask Destination MAC is enabled on -
No Slots
Mask Vlan is enabled on -
No Slots
Mask Inner-Vlan is enabled on -
No Slots
Mask ISID is enabled on
-
```

Use the **show load-balance mask-option pbb** command to view the pbb mask sub-options.

```
device# show load-balance mask-options pbb
Mask PBB options -
Mask PBB Customer L2 Header is enabled on -
All Slots
Mask PBB Customer IPv4/IPv6 Header is enabled on -
Slot 1
Slot 2 -   NPID 1
```

Sample configurations

You can configure a dual tagged and an ISID endpoint on the same port in different VPLS instances as shown below.

```
device(config)# router mpls
device(config-mpls)# vpls test5 105
device(config-mpls-vpls-test5)# vc-mode tagged
device(config-mpls-vpls-test5)# vlan 105 isid 1050
device(config-mpls-vpls-test5-vlan-105-isid-1050)# tag e 2/1
device(config-mpls-vpls-test5-vlan-105-isid-1050)# vpls test6 106
device(config-mpls-vpls-test6)# vlan 106 inner-vlan 1060
device(config-mpls-vpls-test6-vlan-106-inner-vlan-1060)# tag e 2/1
device(config-mpls-vpls-test6-vlan-106-inner-vlan-1060)# vpls test7 107
device(config-mpls-vpls-test7)# vc-mode tagged
device(config-mpls-vpls-test7)# vlan 106 isid 1060
device(config-mpls-vpls-test7-vlan-106-isid-1060)# tag e 2/1
device(config-mpls-vpls-test7-vlan-106-isid-1060)#
```

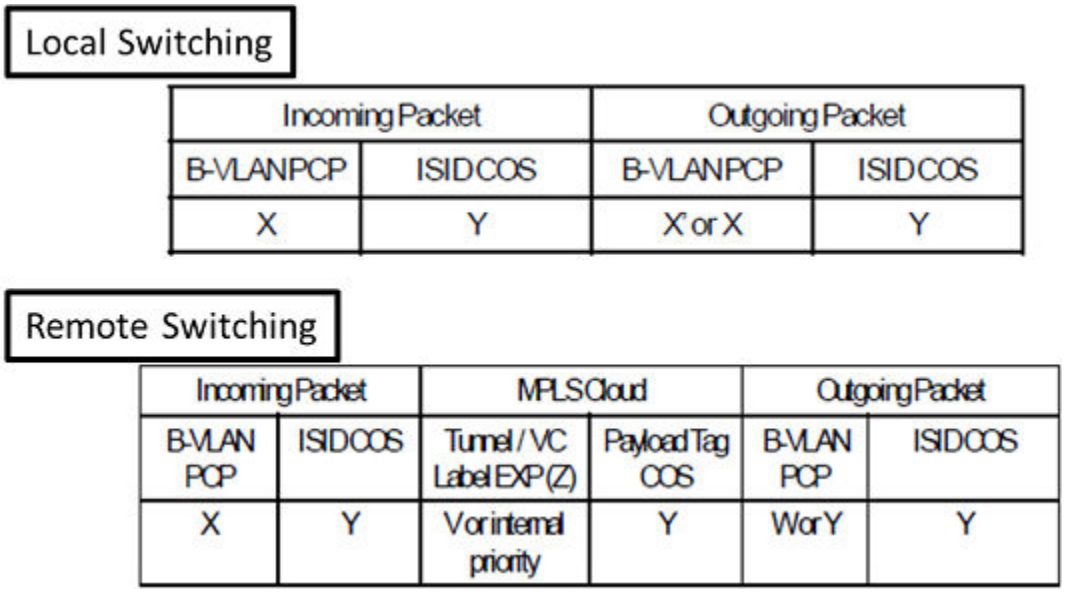
You can have two or more ISID endpoints in the same VPLS instance with different BVIDs on different ports as shown below.

```
device(config)# router mpls
device(config-mpls)# vpls test8 108
device(config-mpls-vpls-test8)# vc-mode tagged
device(config-mpls-vpls-test8)# vlan 108 isid 1080
device(config-mpls-vpls-test8-vlan-108-isid-1080)# tag e 2/1
device(config-mpls-vpls-test8-vlan-108-isid-1080)# vlan 109 isid 1080
device(config-mpls-vpls-test8-vlan-109-isid-1080)# tag e 2/2
device(config-mpls-vpls-test8-vlan-109-isid-1080)#
```

CoS with ISID to ISID endpoints

Figure 45 illustrates the behavior when using CoS with ISID to ISID endpoints.

FIGURE 43 CoS Treatment



LEGEND:

X - original B-VLAN PCP.

Y - original ISID COS.

X' - mapped PCP bits from internal priority (X contributes to internal priority) using PCP encode table.

V - mapped EXP bits from internal priority (X contributes to internal priority) using EXP encode table.

Z - incoming EXP bits as described by Tunnel/VC label column = V or internal priority.

W - mapped PCP from internal priority (Z contributes to internal priority) using PCP encode table.

The 'or' option in the Tunnel/VC label column is to differentiate when 'qos exp encode policy' is on (default) or off.

The 'or' option in the Outgoing B-VLAN column is to differentiate when 'qos pcp encode policy' is on (default) or off.

Local switching

The following configuration guidelines should be considered for local switching when using CoS with ISID to ISID endpoints.

- The Internal priority is mapped from the outer VLAN CoS in the incoming packet or incoming port's priority using the decode map.
- The outgoing outer VLAN CoS is mapped from internal priority using egress encoding map by default. The internal priority does not affect the outgoing ISID CoS.
- The outgoing outer VLAN CoS is same as the incoming packet's outer VLAN CoS if the **qos pcp encode-policy off** command is configured on the outgoing interface.
- The outgoing ISID CoS is the same as incoming packet's ISID CoS. This cannot be changed by any configuration.

Remote switching

The following configuration guidelines should be considered for remote switching when using CoS with ISID endpoints.

Local end point to remote peer (MPLS cloud)

The following configuration guidelines should be considered for remote switching when using CoS with ISID endpoints to a MPLS Cloud.

- The internal priority is mapped from outer VLAN CoS in the incoming packet or incoming port priority using the decode map.
- If VPLS or LSP CoS is configured, this value overrides internal priority.
- The outgoing tunnel or VC label EXP bits are mapped from internal priority using the egress encoding map by default.
- The outgoing tunnel or VC label EXP bits are set to the internal priority if the **qos exp encode-policy off** command is configured on the outgoing interface.
- The CoS of the payload tag is the same as the incoming packet ISID CoS and cannot be overridden by any other configuration.

Remote peer (MPLS cloud) to local end point

The following configuration guidelines should be considered for remote switching when using CoS with a MPLS Cloud to ISID endpoint.

- The internal priority is mapped from the tunnel or VC label EXP bits in the incoming packet or incoming port priority using the decode map.
- VPLS CoS will not override internal priority.
- The outgoing outer VLAN CoS is mapped from internal priority using egress encoding map by default. The internal priority does not affect the outgoing inner VLAN CoS.
- The outgoing outer VLAN CoS is the CoS in the payload tag if the **qos pcp encode-policy off** command is configured on the outgoing interface.
- The outgoing ISID CoS is the CoS in the payload tag.

End-to-end behavior

The following configuration guidelines should be considered for remote switching when using end-to-end behavior for CoS.

- The incoming outer VLAN CoS on the ingress router can affect the outgoing outer VLAN CoS on the egress router by default. This can be overridden by the VPLS or LSP CoS.
- The incoming ISID CoS on the ingress router is preserved in the outgoing outer VLAN COoS if the **qos pcp encode-policy off** command is configured on the outgoing interface of the egress router.
- The incoming ISID CoS on the ingress router is always preserved in the outgoing ISID COS of the egress router.

Configuration mismatch - forwarding behavior

The same ISID value must be configured on ingress and egress devices. Typically, when the MPLS egress device receives the first packet, the CAMs are programmed so that forwarding can be done in hardware. If there is a configuration mismatch, if the MPLS egress device has received a packet with an ISID that is not the same as the configured ISID, the software can detect the mismatch and not program the CAMs. This will cause the packets to be discarded.

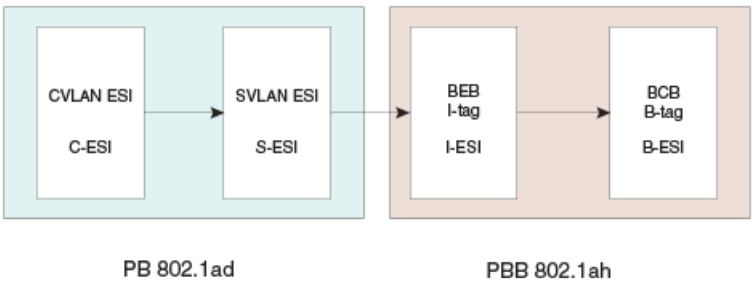
> NOTE
> If the first packet arrived with the right ISID, the CAMs will be programmed as expected. Subsequently, if the packet arrives with an incorrect ISID, it will not be possible for the hardware to detect the mismatch and packets will be forwarded to the VPLS endpoint with the incorrect ISID.

When CPU protection is enabled, broadcast or unknown unicast packets will continue to be forwarded even when they arrive with an incorrect ISID.

Adding and removing VLANs and ESIs

Use Figure 46 and the tables in this section for information about adding and removing VLANs and ESIs. Refer to the Figure 46chapter for additional information.

FIGURE 44 Association of ESI and VLAN for different stages



Adding a VLAN to an ESI

TABLE 47 Adding a VLAN to an ESI

Elementn	ESI encapsulation	Configuration	Duplicates	Decision
Add CVLAN	CVLAN		• No duplicate within same ESI • No duplicates among other client ESIs of this ESI's parent	OK
Add SVLAN	SVLAN	No client ESI defined, No I-ESI provider ESI defined	• No duplicate within same ESI • No duplicates across all provider ESIs	OK
		No client ESI defined, No I-ESI provider ESI defined	• Duplicates among all S-ESI with no provider ESI	NOT OK
		I-ESI provider ESI defined	Duplicates among all S-ESI belonging to the I-ESI	NOT OK
Add SVLAN	SVLAN	Client ESI defined		
		Number of SVLANs in the ESI == 0	• No duplicates across all provider ESIs	OK
		Number of SVLANS in the ESI >= 1		NOT OK

Adding a source ESI to a target ESI

TABLE 48 Adding a source ESI to a target ESI

	Source ESI encapsulation	Target ESI encapsulation	Condition	Duplicates check	Decision
Add ESI (PB)	CVLAN	CVLAN			NOT OK
	CVLAN	SVLAN	Number of SVLANS in the ESI <= 11	1. No duplicates across CVLANs 2. No duplicates among other client ESIs	OK
	SVLAN	SVLAN			NOT OK
	SVLAN	CVLAN			NOT OK
PBB	SVLAN	ISID	Number of ISIDs in the ESI <= 1	Subject to duplicate check	OK
	SVLAN	SVLAN			NOT OK
	SVLAN	BVLAN			NOT OK
	ISID	BVLAN	Number of BVLAN in the ESI <= 1	Subject to duplicate check	OK
	ISID	ISID			NOT OK
	ISID	SVLAN			NOT OK

Deleting a VLAN

TABLE 49 Deleting a VLAN

VLAN type (ESI-encapsulation)	Actions
CVLAN	Delete CVLAN from ESI
SVLAN	Delete SVLAN from ESI
BVLAN	Delete BVLAN from ESI
ISID	Delete ISID from ESI

Deleting an ESI

TABLE 50 Deleting an ESI

	ESI-encapsulation	Actions
PB	CVLAN	• Remove ESI from any associated provider ESI's client ESI list • Delete all VLANs bound to the ESI • Delete ESI and return to free pool
PB	SVLAN	• Delete SVLANs bound to the ESI • Remove link with any client ESI and sever links of all client ESIs • Remove ESI from its provider ESI's (such as PBB) client ESI list • Delete ESI and return to free pool

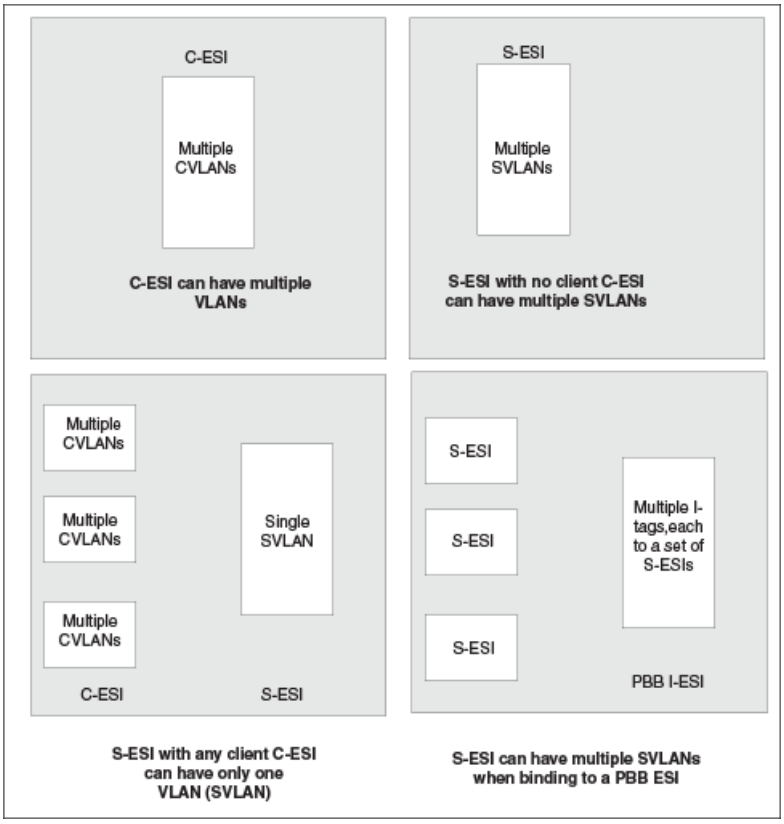
TABLE 50 Deleting an ESI (continued)

	ESI-encapsulation	Actions
PBB	SVLAN	• Remove ESI from any associated provider ESI's client ESI list • Delete all VLANs bound to the ESI • Delete ESI and return to free pool
PBB	ISID	• Delete ISID bound to the ESI • Remove link with any client ESI and sever links of all client ESIs • Remove ESI from its provider ESI's (such as PBB) client ESI list • Delete ESI and return to free pool
PBB	BVLAN	• Delete BVLANs bound to the ESI • Remove link with any client ESI and sever links of all client ESIs • Remove ESI from its provider ESI's (such as PBB) client ESI list • Delete ESI and return to free pool

Valid ESI configuration and interconnection modes

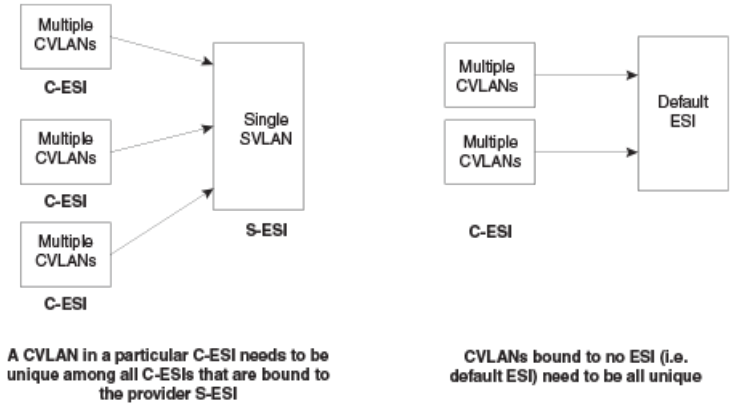
Figure 47 shows the allowable configurations for the ESI elements.

FIGURE 45 Allowable configurations for different ESI elements



Uniqueness requirements for VLANs

FIGURE 46 CVLAN uniqueness requirement



A CVLAN value to be added to a C-ESI must be unique:

- Among all CVLANs within the C-ESI
- If the C-ESI has a provider S-ESI (with or without a SVLAN), the CVLAN needs to be unique across all C-ESI that are clients of the provider S-ESI.

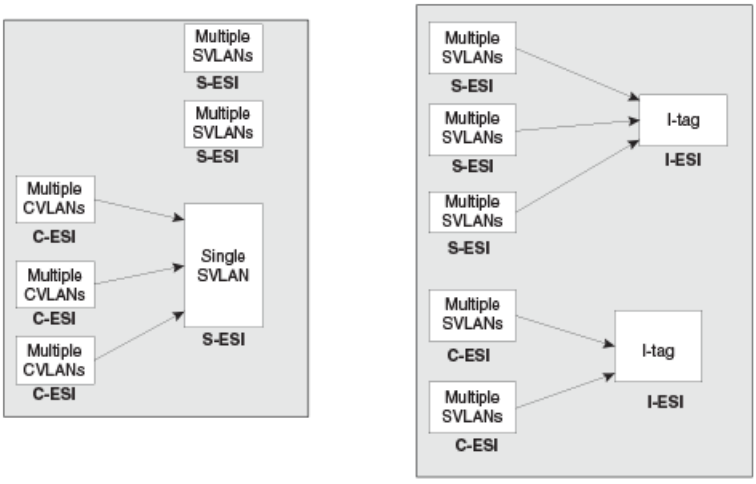
Default ESI

If CVLANs are not bound to a particular C-ESI, that is not entered with any ESI name they are assigned to a default ESI.

- CVLANs need to be unique among all CVLANs that are not bound to a particular C-ESI.

SVLAN uniqueness

FIGURE 47 SVLAN uniqueness for IEEE 802.1ad and IEEE 802.1ah configurations



IEEE 802.1ad (PB)

For IEEE 802.1ad (PB), because the provider only has 4K of SVLANs, the SVLANs must be unique among all S-ESIs

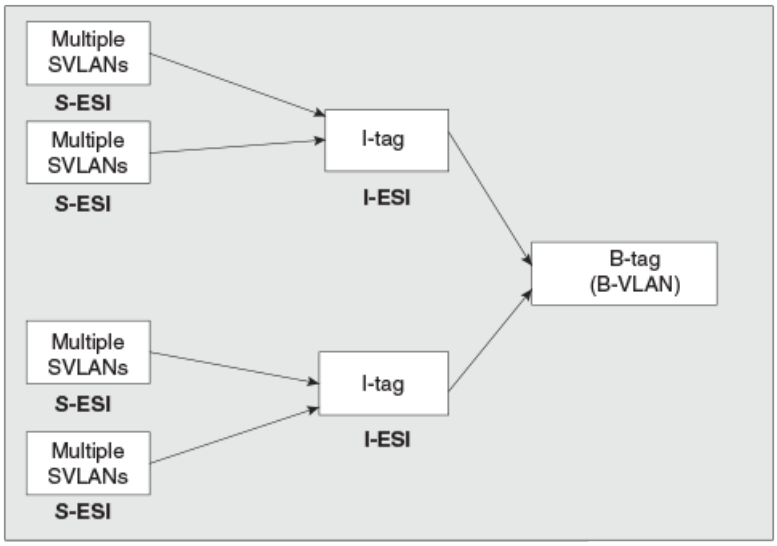
IEEE 802.1ah (PBB)

With IEEE 802.1ah (PBB), multiple SVLANs are mapped to I-tags. These SVLANs may belong to different providers and should be unique only among the SVLANs for the S-ESI to which the SVLAN belongs, and among all the S-ESIs which are clients for the I-ESI for the I-tag encapsulation.

I-tag uniqueness

I-tags must be unique across all I-ESIs that are associated with a single B-ESI provider ESI, as shown in Figure 50.

FIGURE 48 I-tag mappings



Provider Backbone Bridging (PBB) Networks for the Brocade NetIron XMR Series and the Brocade NetIron MLX Series

Overview

The IEEE 802.1ah Provider Backbone Bridges (PBB) standard was developed to address the limitations of Provider Bridges (PB) and to add additional capabilities sought by Service Providers. When compared to a PB network, a PBB network deployment offers simplified operations, lower capital expenditures, and overall better scalability in terms of the number of supported customers. This section provides an overview of PBB for Brocade NetIron XMR Series and Brocade NetIron MLX Series, describing its advantages and examines common PBB deployment scenarios.

Provider Backbone Bridges

The Provider Backbone Bridges (PBB) standard, (IEEE 802.1ah), was developed to address the limitations of the Provider Bridges (PB) standard, (IEEE 802.1ad), and to add additional capabilities sought by Service Providers.

PB allows Service Providers to use a V-LAN identifier (VID) space separate from the customer VID (C-VID) space. PB adds a Service Provider VLAN Tag (S-TAG) containing a Service Provider VID (S-VID) to Ethernet frames (Figure 51). Because PB stacks a second VLAN tag to Ethernet frames, it is also known as "Q-in-Q," as a reference to the standard that originally defined VLAN tags, that is, IEEE 802.1Q, which is known as defining "Q" frames.

The S-VID field of the S-TAG is 12 bits long, which is the same length of a C-VID field of a customer VLAN Tag (C-TAG). Even though 12 bits can address up to 4096 distinct values, two values have special meaning and are reserved. Therefore, the Service Provider is limited to at most 4090 distinct S-VID values to identify service instances, that is, services or customers in a PB network. Another drawback is that PB frames are addressed by customer Media Access Control (MAC) addresses. This means that core Ethernet switches in a PB network have to learn all the source MAC addresses of all the customer frames traversing the core of the PB network. Thus, the size of the MAC address tables of core PB switches ultimately limits the number of customers that can be supported by a PB network.

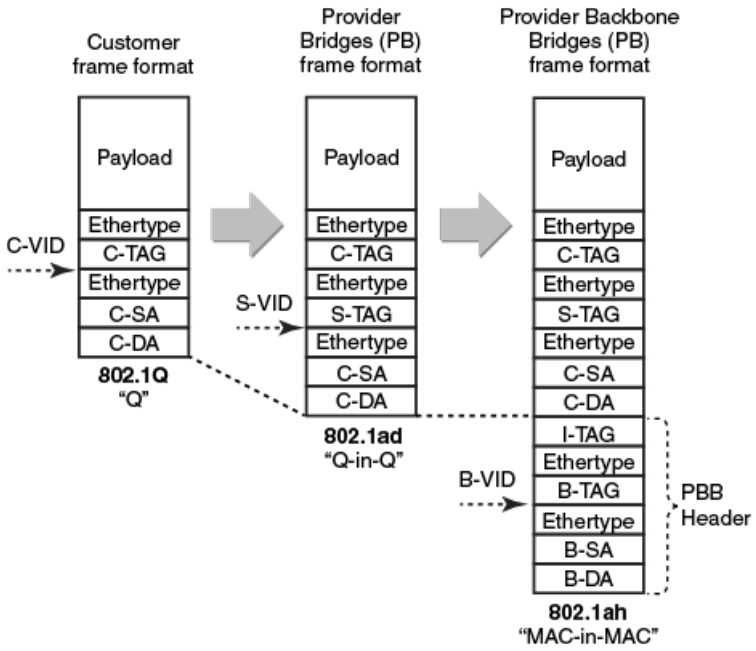
To address the above described PB shortcomings, PBB adds a hierarchy view to Ethernet by encapsulating PB frames with a PBB header (which becomes the equivalent of a "Service Provider MAC header") containing a Backbone Destination MAC Address (B-DA), Backbone Source MAC Address (B-SA), and two new tags (Figure 51), which are described later in this document. What makes the B-DA and B-SA "backbone" addresses is the fact that these are MAC addresses of Service Provider's PBB edge switches. An edge PBB switch encapsulates an ingress PB frame with a PBB header containing the destination MAC address of an appropriate egress edge PBB switch. The egress edge PBB switch removes the PBB header and forwards the frame to an attached PB network. Because PBB adds a PBB header containing new destination and source MAC addresses, it is also known as "MAC-in-MAC."

By adding the PBB header, PBB isolates the Service Provider and customer address spaces. This means that Ethernet switches in the core of the Service Provider network will no longer learn customer MAC addresses or use customer MAC addresses to forward

customer frames to their destinations. This improves the scaling of the Service Provider network in terms of the number of supported customers, since the number of supported customers is no longer directly tied to the size of the MAC address tables of the core Ethernet switches. In addition, the Service Provider network is now protected from customer network failures, since frame forwarding is now based on its own PBB header. Moreover, customers benefit from added security, since the customer's MAC addresses are no longer learned or used for frame forwarding decisions in the core of the Service Provider network.

As additional benefits to the Service Provider, PBB has the potential to simplify operations, e.g., by separating the customer and Service Provider addressing spaces, and to lower capital expenditures by reducing the cost of Ethernet switches used in the core of the network, since memory and processing power requirements are reduced by limiting MAC address learning to backbone MAC addresses.

FIGURE 49 Customer, PB, and PBB frame formats



The Backbone Service Instance Tag (I-TAG) contains a Backbone Service Instance Identifier (I-SID), which is 24 bits long. The I-SID field allows a Service Provider to identify up to 2 to the power of 24, that is, over 16 million, service instances. In other words, over 16 million services or customers can be uniquely identified using the I-SID field. Therefore, PBB's I-TAG allows for highly scalable services by eliminating the 4090 service instances limitation of PB.

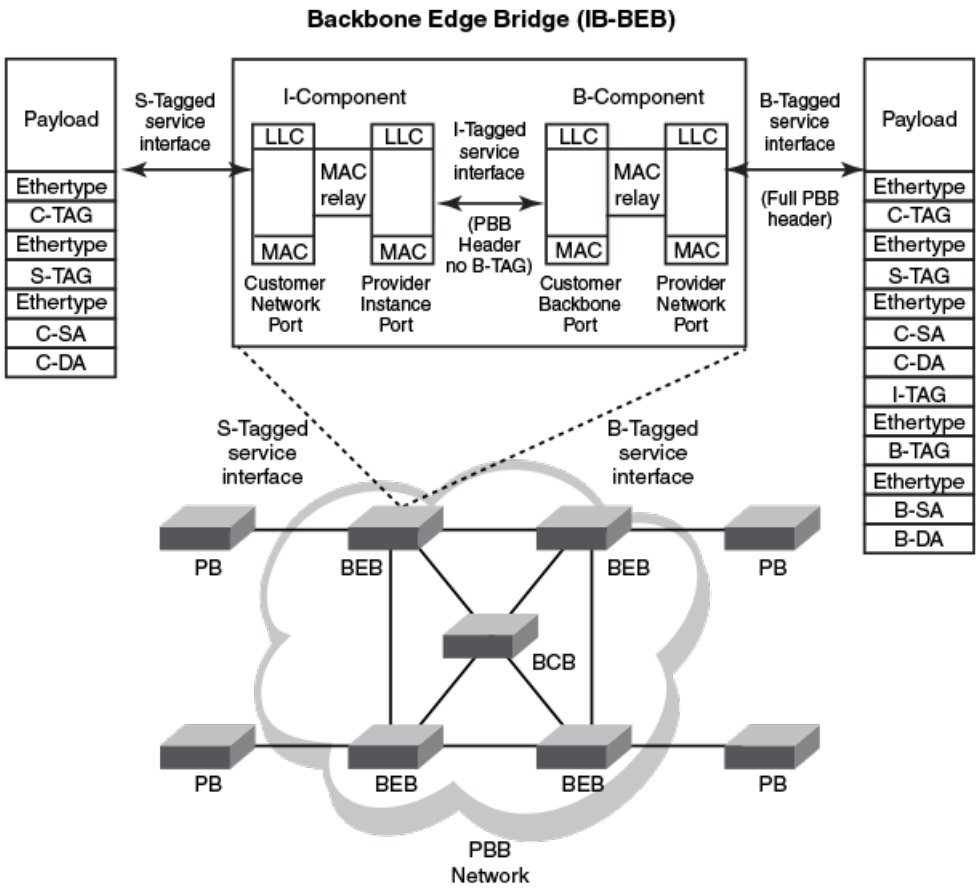
The semantics and the structure of the Backbone VLAN Tag (B-TAG) are identical to that of the PB S-TAG. The B-TAG was designed this way so that core PBB switches do not need to be aware of PBB. In fact, standard PB switches can be used in the core of a PBB network. Only the switches at the edge of the Service Provider PBB network need to be aware PBB.

A PBB network uses two types of bridges (Backbone Edge Bridge (BEB) operation on page 295): Backbone Edge Bridges (BEB) and Backbone Core Bridges (BCB). As explained above, the functionality required from a BCB is the same as a standard IEEE 802.1ad PB bridge. A BEB is used at the boundary of a PBB network to add and remove the PBB header.

Backbone Edge Bridge (BEB) operation

A BEB containing an I-Component and a B-Component is called an IB-BEB. The B-Component of an IB-BEB forwards packets towards the PBB network based on backbone MAC addresses (that is, it learns backbone MAC addresses), while the I-Component forwards packets towards the PB network based on the customer MAC addresses (that is, it learns customer MAC addresses). Brocade NetIron XMR Series and Brocade NetIron MLX Series only support IB-BEB. Neither support just the I-component BEB or just the B-component BEB.

FIGURE 50 Backbone Edge Bridge operation



Service instance

A Service Instance (SI) is also known as a bridging domain. It is defined as a single flooding domain where any traffic that requires flooding is always flooded to all endpoints under the same SI. When an unknown destination packet is received at a PBB endpoint it will cause flooding of this packet to all the endpoints (source port suppressed). The following types of endpoints defined here are supported by PBB.

Untagged endpoint

By default a port is configured with a Ethernet type (TPID) of 0x8100. If an untagged packet is received by the port, it is accepted and switched within the PBB instance to the desired destination MAC. If a 0x8100 tagged packet is received and the port is operating as untagged mode with the default TPID of 0x8100, the packet will be dropped as an invalid packet.

Port-based untagged endpoint

Port-based untagged endpoint is defined to always accept whatever packet is received by the given port whether there is any tag present in the packet received. As explained earlier, if the packet received on an untagged mode interface where the received packet contains the tag that matches the configured port Ethernet type value, it is dropped as an invalid packet. To achieve the desired behavior of treating the entire packet as an untagged packet (no tag stripping/insertion), the user must configure the Ethernet type of the port to a value that will not match the Ethernet type of any packets sent to this port. This way, although the packet received may contain a tag its value will never match to the configured TPID value of the port (avoid setting it to a known protocol Ethernet type value) thus the tag will be treated as part of the customer payload. A suggested value of 0x9FFF can be used for this purpose as this value is not listed in the IEEE Ethernet Type Listing as a reserved value for a known protocol at the time of this document creation. Please always refer to IEEE Ethernet Type field public listing to avoid setting to a value that may conflict with a known protocol.

C-Tagged endpoint

C-tagged endpoint is defined as an endpoint that considers all packets received with the top most tag being the Customer Tag (C-Tag). This is achieved by configuring a tagged endpoint under the PBB instance. The user must configure the Ethernet type of the port to match the C-Tag TPID value of the received packets. Packets sent out through the C-tagged endpoint will have a C-tag with C-VID 100 added at egress to the endpoint.

Example 1:

If the packet received has a C-VID of 100 and the destination is another C-Tagged endpoint with C-VID 200, when the packet exits the destination C-Tagged endpoint, it will have the C-VID of 200. The original C-VID 100 is "translated" in this case.

Example 2:

If the packet is destined to a port based untagged endpoint, the packet will be sent out without the original C-VID 100 tag because it was removed at the ingress of the C-tagged endpoint.

Example 3:

If the packet is destined to an IB-Tagged endpoint, the original tag that contains the C-VID of 100 is stripped at ingress and a PBB header (contains the B-MACs, B-Tag and I-Tag) is inserted when the packet exits through the IB-Tagged endpoint. Optionally, if S-VLAN keep mode is configured, an S-Tag will also be inserted. More details about S-VLAN keep mode is discussed in a later section.

S-Tagged endpoint

S-Tagged endpoint is defined as an endpoint that considers all packets received with the top most tag being the Service provider Tag (S-Tag). The user must configure the Ethernet Type value of the port to match the S-Tag TPID value of the packets received on the specified port. The packet processing operation whether it is S-Tagged or C-Tagged is equivalent. When flooding is performed, a valid packet received through the S-Tagged endpoint will be flooded to all endpoints configured under the same SI, regardless of the egress endpoint tag type.

Dual-tagged endpoint

Dual-tagged endpoint is defined as an endpoint that expects two tags (Q-in-Q) where the top most tag must match the configured Ethernet Type value of the port and the inner tag must match the default TPID of 0x8100. The dual-tagged endpoint configuration is NOT supported with PBB on Brocade NetIron XMR Series and Brocade NetIron MLX Series.

IB-Tagged Endpoint

An IB-Tagged endpoint represents all frames on a physical port with a particular B-VID and a particular I-SID. When such a packet is received from the IB-endpoint destined to another local endpoint of the same PBB instance, the PBB header will be stripped and a destination tag may be inserted accordingly based on the exiting endpoint configuration.

Example 1:

If the packet is destined to a C-Tagged endpoint with C-VID 100, the original PBB header is stripped and a C-Tag with C-VID 100 is inserted as the packet exits the C-tagged endpoint. The TPID value will be configured for the Ethernet type value of the exit port.

Example 2:

If the packet is destined to an S-Tagged endpoint with S-VID 300, the original PBB header is stripped and an S-Tag with S-VID 300 is inserted as the packet exit the S-tagged endpoint. The TPID value will be configured for the Ethernet type value of the exit port.

Example 3:

If the packet is destined to a port based untagged endpoint, the original PBB header is stripped. No additional tag will be inserted as the packet exit the untagged endpoint.

Example 4:

If the packet is destined to another IB-Tagged endpoint with the same B-VID and ISID value of the received PBB frame, the original PBB frame is NOT touched (the PBB header remains unchanged) and will be send out towards the specified outgoing interface as determined by the BEB's switching logic.

IB-Tagged Endpoint with S-VLAN Keep mode

The system can be configured with S-VLAN keep mode which indicates that packets received at the IB-Tagged endpoint from the PBB network will contain an S-Tag after the I-Tag field (see Backbone Edge Bridge (BEB) operation on page 295). When such a packet is destined to a local endpoint, in addition to stripping the PBB header the S-Tag is also removed.

Example 1:

If the packet is destined to a C-Tagged endpoint with C-VID 100, the original PBB header and the S-Tag is stripped and a C-Tag with C-VID 100 is inserted as the packet exit the C-tagged endpoint. The TPID value will be what is configured for the exiting port's Ethernet type value.

Example 2:

If the packet is destined to an S-Tagged endpoint with S-VID 300, the original PBB header and the S-Tag is stripped and an S-Tag with S-VID 300 is inserted as the packet exit the S-tagged endpoint. The TPID value will be what was configured for the exiting port's Ethernet type value. The S-VID of the S-Tag in the original PBB frame is "translated" into the S-VID of the destination S-Tagged endpoint. The PCP/DEI is preserved or modified depends on the PCP encode and decode setting of the receiving/destination ports.

Example 3:

If the packet is destined to another IB-Tagged endpoint with the same B-VID and ISID value of the received PBB frame, the original PBB frame is NOT touched (the PBB header and S-Tag remains unchanged) and will be send out towards the specified outgoing interface as determined by the BEB's switching logic.

S-VLAN VID Value setting when packet exit IB-Tagged Endpoint with S-VLAN Keep mode

If a packet is received from a C-Tagged or S-Tagged endpoint destined towards an IB-Tagged endpoint under S-VLAN keep mode, the S-VLAN VID value will depend on the original VLAN ID received. If the packet was received from an untagged endpoint, then the untagged VLAN value of the port will be used as the S-VLAN VID as it goes out of the IB-Tagged endpoint.

Example 1

If the packet is received from a C-Tagged endpoint with C-VID 100 and destined to an IB-tagged endpoint, the original tag that contains the C-VID of 100 is stripped and a PBB header (contains the B-MACs, B-Tag, I-Tag, and S-Tag) is inserted when the packet exits through the IB-Tagged endpoint. The S-Tag S-VLAN VID in the PBB header will have a value of 100.

Example 2

If the packet is received from a S-Tagged endpoint with S-VID 200 and destined to an IB-tagged endpoint, the original tag that contains the S-VID of 200 is stripped and a PBB header (contains the B-Macs, B-Tag, I-Tag, and S-Tag) is inserted when the packet exits through the IB-Tagged endpoint. The S-Tag S-VLAN VID in the PBB header will have a value of 200.

Example 3

If the packet is received from an Untagged endpoint with default port VLAN set as 150 and destined to a IB-tagged endpoint, a PBB header (contains the B-Macs, B-Tag, I-Tag, and S-Tag) is inserted when the packet exits through the IB-Tagged endpoint. The S-Tag S-VLAN VID in the PBB header will have a value of 150.

Customer to ISID mapping

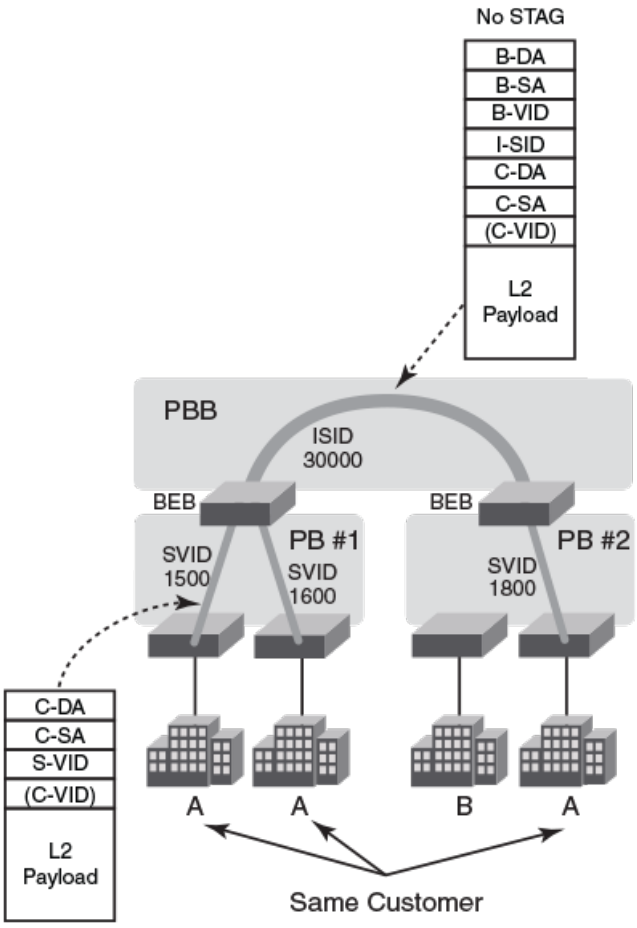
There are two types of Customer to ISID Mapping. 1:1 Customer to ISID Mapping and N:1 Customer to ISID Mapping. Brocade NetIron XMR Series and Brocade NetIron MLX Series only support one single flooding domain per SI regardless of which mapping is chosen. 1:1 Customer to ISID mapping is supported by default. Brocade NetIron XMR Series and Brocade NetIron MLX Series will also support S-VLAN Keep Mode to inter-operate with Brocade NetIron CES Series and Brocade NetIron CER Series and other vendors on the same PBB network that supports the N:1 Customer to ISID mapping.

Regardless of which type of mapping is chosen, Brocade NetIron XMR Series and Brocade NetIron MLX Series will always strip the incoming S-Tag and add the S/C-Tag as it goes out on the BEB's S/C-Tag endpoint based on what is configured and not based on what is received. Optionally the S-Tag TPID value can also be "translated" based on the exiting port's configured Ethernet Type value.

1:1 Customer to ISID mapping

With 1:1 customer to ISID mapping, a single customer is mapped to an ISID value. There may be many S-Tag endpoints configured under the same SI, but they are all considered as belonging to one customer. The BEB strips the S-Tag at the ingress of the BEB and inserts the S-Tag at the egress of the BEB. There is no S-TAG in the PBB frame when it traverses the PBB network. Any PBB instance configured in the system is considered as one SI which implies one single flooding domain. 1:1 Customer to ISID mapping shows an example of a 1:1 customer to ISID mapping.

FIGURE 51 1:1 customer to ISID mapping



N:1 Customer to ISID Mapping Interop (S-VLAN Keep Mode)

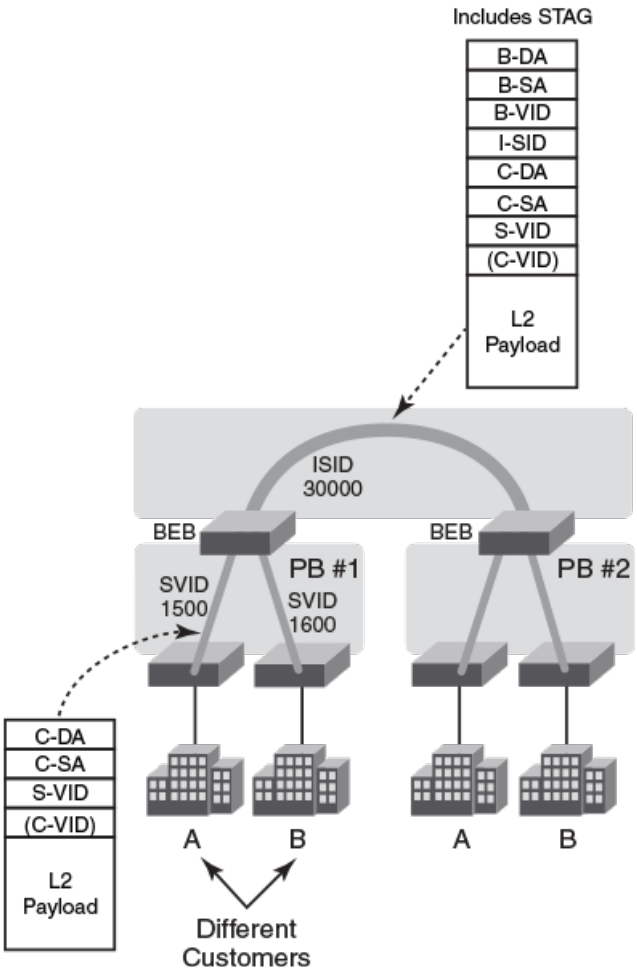
The benefits of N:1 customer to ISID mapping include the following:

- Multiple customers are mapped to the same SI to use the same ISID.
- The S-VLAN at the ingress is carried all the way to the egress side of the PBB network.
- When flooding of a packet is required, it only floods to the endpoints that have the same S-VID.

Brocade NetIron XMR Series and Brocade NetIron MLX Series supports S-VLAN Keep Mode. This deviates from true N:1 Customer to ISID mapping in that it will NOT have different flooding domains based on the S-VLAN, but continue to use one single flooding domain per VPLS instance. When flooding of packets is required, it will always flood to all the endpoints (source port suppressed) within the same VPLS instance (SI).

If the S-VLAN keep mode is desired, you must ensure that all the Brocade NetIron XMR Series and Brocade NetIron MLX Series nodes on the same PBB network are all configured with the S-VLAN keep mode. Once a BEB is configured with S-VLAN keep mode any PBB frames received without the proper S-Tag present in the PBB frame will be treated as invalid packet and discarded.

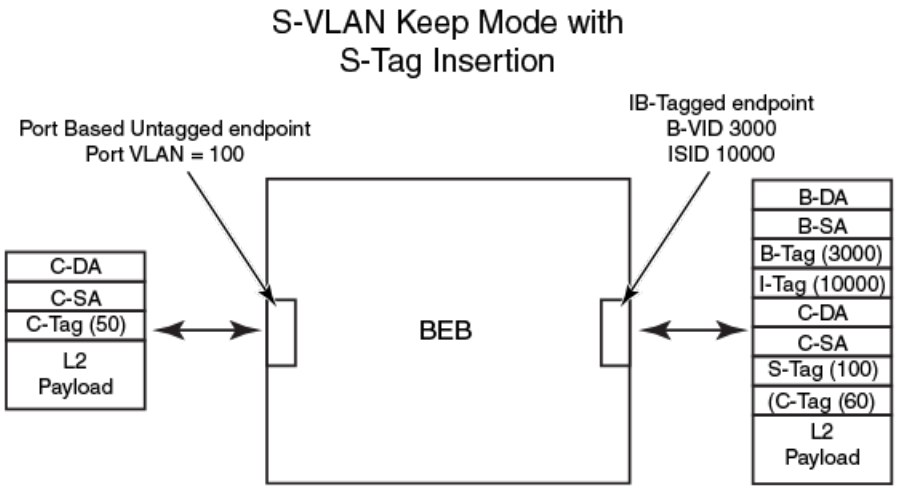
FIGURE 52 S-VLAN keep mode



Port Based Untagged Frame with S-Tag Insertion

With S-VLAN keep mode it is also possible to insert an S-Tag on top of whatever tag(s) the original packet may already have when it goes out into the PBB network. This is done by configuring a port-based untagged endpoint with the desired S-VID as the Port's default VID and send port based untagged packets into this port. When the corresponding packet (shown in Figure 55 as a C-Tagged packet) is switched towards an IB-endpoint into the PBB network, the BEB will insert the PBB header as well as inserting an S-Tag with the S-VID set to the ingress port's configured VLAN value. The S-Tag TPID will be set to whatever was configured as the system-wise S-Tag Ether Type value.

FIGURE 53 S-VLAN keep mode with S-Tag insertion



PBB packet switching

There are several scenarios on how a packet is switched within the BEB.

- Packet may be switched between a local endpoint and an IB endpoint of the SI.
- Packet may be switched between two local endpoints of the SI.
- Packet may be switched between two IB endpoints where the BEB is acting as a BCB.
- Unknown C-DA packets flooding handling.

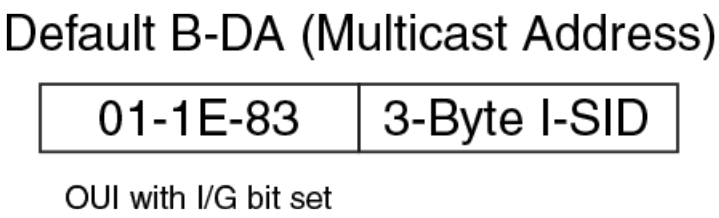
PBB Packet Received at the IB endpoint

When the sender does not know where the C-DA is located, the packet is flooded with the configured flooding B-DA over the B-VLAN configured for the respective SI. If the sender knows which BEB owns the intended C-DA (C-DA was learnt and associated with a particular BEB's B-MAC), it will send the packet to that destination BEB.

Unknown Destination PBB Packet Handling

When a BEB receives a PBB packet with the default backbone multicast destination address as its B-DA and the corresponding ISID is of interest (it has configuration of SI that has matching BVLAN and ISID), it will examine the packet to determine if the associated C-DA in the PBB packet is a known MAC address. If it is known, it will forward the packet to the port that learned the C-MAC. If the C-DA is also unknown to the BEB that received this PBB packet, it will need to flood this packet to all its local endpoints as well as to all the IB-endpoints of the corresponding SI. When such flooding occurs, the packet destined towards the PBB networks will retain the original received PBB header without any modification. The default backbone multicast destination MAC address is constructed by concatenating the three octet OUI 00-1E-83 with the three octet I-SID (and asserting the I/G bit to signify a group MAC address). The resulting B-DA is shown in Figure 56.

FIGURE 54 Unknown Destination PBB Packet Handling



If the PBB packet received was for an ISID that the BEB does not care about (no SI configured that matches the B-VLAN and ISID value of the PBB packet), then the packet is forwarded based on the B-Tag and the B-MACs as a regular Layer 2 packet. No attempt will be made to look into the inner MAC of the PBB packet.

NOTE
No validation is made between the 3-byte I-SID value within the default B-DA multicast address to the PBB header's ISID value. Brocade NetIron XMR Series and Brocade NetIron MLX Series will always use the PBB header ISID value when a switching decision is made.

Known Destination PBB Packet handling

When ingress BEB receives a packet with known C-DA that it learnt from one of the IB-endpoint, it will forward the received packet towards the IB-endpoint that it learnt the C-DA. The PBB header will contain the B-VID and ISID that correspond to what was configured for the corresponding SI. The B-DA will be the B-MAC associated with the learnt C-DA. The B-SA will be set to the chassis base MAC address.

Unknown Destination PB packet Handling

When a packet is received at a local endpoint (S/C-Tagged or Port based untagged), if the C-DA is unknown, it will also require flooding of this packet towards all local endpoints as well as towards the PBB network via the IB-endpoints.

If a flooding B-DA was configured for this SI, when it floods the packet to the IB-endpoints, it will use the configured flooding B-DA. Otherwise, it will use the default backbone multicast destination MAC address when it floods the packet towards the PBB network.

BEB Acting as BCB

When a BEB receives a PBB packet with the B-DA and it is not its own chassis MAC, then the packet must be destined for some other PBB node within the PBB network. In this case, the BEB will act as a BCB and Layer 2 switch the received packet based on the B-MACs and the B-VLAN towards the next PBB node. The ISID portion of the packet is ignored and treated as part of the payload.

PBB MAC Learning

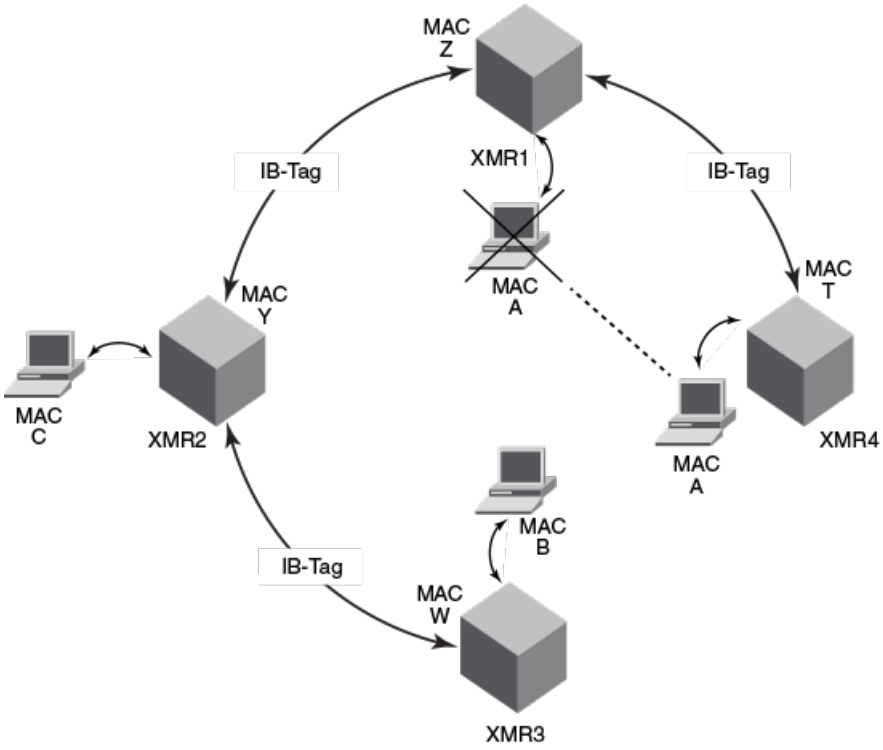
With the introduction of PBB, there will be some MAC interactions for the B-MACs involved between Layer 2 (for B-VLAN) and PBB.

MAC Learning for PBB Packets

PBB packets are MAC-in-MAC packets. There are two types of MAC learning involved depending on whether the ISID involved in the received PBB packet is of any interest. If there is no such SI configured for the PBB packet's B-VLAN and ISID, only the outer MAC (B-SA) is learned via the regular Layer 2 MAC management for the corresponding B-VID. A regular Layer 2 endpoint must be configured for each B-VLAN used as an IB-endpoint in a PBB instance.

If the PBB packet received has an SI configured that matches the B-VLAN and ISID of the packet's PBB header, the inner MAC (C-SA) will be learned by the corresponding PBB instance and an association is made between the C-SA to the B-SA of the received PBB packet. The B-SA association under PBB is primarily used to program the hardware so that it knows how to set up the PBB header for packets destined to the C-MAC that was associated with a particular B-MAC. The B-DA of such packet forwarding will be set up according to this association. Another usage of this B-SA association under PBB is to enable detection of B-MAC movement where the C-MAC association to B-MAC may have changed from B-MAC-Z to B-MAC-T in Figure 57.

FIGURE 55 MAC Learning for PBB packets



L2 MAC Installment by PBB

When packet is being switched by the hardware based on the inner MACs (C-DA and C-SA), the outer MACs (B-DA and B-SA) will stop hitting the regular Layer 2 CAMs programmed for the associated B-SA. So without any intervention from PBB, these B-MACs once learned by the corresponding Layer 2 B-VLAN, will start to age out. Once aged out, if another PBB packets is received that is being switched based on the outer B-MACs, will end up causing re-learning those B-MACs again and unnecessary flooding even though there may exist a constant traffic for some packet flow that is being switched based on the inner MACs that associate with these aged out B-MACs.

In order to address this Layer 2 B-MAC aging out issue, PBB installs the associated B-MACs to the appropriate Layer 2 VLAN MAC space and marking them similar to static MACs where aging is disabled. These installed B-MACs are dynamically installed based on the C-MAC to B-MAC association. These installed B-MACs will be flush-able by the following methods:

- Explicit CLI command such as "clear mac".
- Topology Changes Notification (TCN) that caused flushing such B-MACs.

Once the B-MAC is cleared by the CLI method or TCN method, the B-MAC will cease to exist in the Layer 2 VLAN space until either PBB re-install it again due to C-MAC relearned and re-associated with the B-MAC or B-MAC is learnt by Layer 2 B-VLAN based on BCB forwarding action.

When the last C-MAC associated with the corresponding B-MAC is aged out, the corresponding B-MAC that was previously installed will be re-programmed to allow aging and if there were no Layer 2 traffic hitting this B-MAC, it will be aged out by Layer 2 naturally.

Temporarily Learning of C-MACs on Transit BEB

Although one of the main purposes of deploying the PBB network is to reduce the learning of C-MACs to a smaller set, the BEB may momentarily learn the C-MACs that does not belong to the BEB due to unknown flooding traffic using the default backbone multicast destination MAC address. Once the destination becomes known, the sender will stop using the default mcast flood MAC and use the learned B-MAC, those C-MACs that were temporarily learned on the transit BEB will start to age out.

PBB PCP/DEI Setting

Behaviors of the B-Tag PCP/DEI, I-Tag PCP/DEI and S-Tag PCP/DEI settings are described in the following sections.

B-Tag PCP Setting

The B-Tag PCP setting has two options:

1. Encode PCP On - This is the default case where the B-Tag PCP is derived based on the internal priority modified by the selected PCP encode table.
2. Encode PCP Off - When this is configured on the egress IB endpoint interface, the B-Tag PCP value will be set to a fixed value depends on the "Forced-PBB-PCP" configuration on the corresponding B-VLAN:
   a) **Forced-PBB-PCP** is set on B-VLAN - This allows user to force the B-Tag PCP setting. The PBB header B-Tag PCP value will be set to the specified forced-PBB-PCP value.
   b) **Forced-PBB-PCP** is NOT configured for the B-VLAN - This will cause the PBB header B-Tag PCP value set to "0".

NOTE

If a packet ingress through a C-Tagged endpoint is destined to a remote PBB node, the C-Tag PCP value will be used the same way as described above.The Internal priority setting is based on the original packet's PCP value operated by the port's PCP decode table, as explained in the *Brocade NetIron QoS and Traffic Management Configuration Guide* .The **forced-PBB-PCP** configuration option has no effect on the regular Layer 2 traffic using the same B-VID. For pass through PBB packets (BEB acting as a BCB case), the B-Tag PCP value is not modified at all. The existing "priority" configuration on the B-VLAN operates independently from the **forced-PBB-PCP** . It governs on the PCP setting of the regular Layer 2 traffic as well as pass through PBB traffic.

B-Tag DEI Setting

The B-Tag DEI setting has the following options:

1. Encode PCP On - This is the default case where the B-Tag DEI is derived based on the internal drop precedence and the **qos use-dei** setting configured on the egress IB interface.
   a) "qos use-dei" On - The B-Tag DEI setting is based on the internal drop precedence set up at the ingress.
   b) "qos use-dei" Off - The B-Tag DEI setting is set to "0" regardless of what the internal drop precedence may be.
2. Encode PCP Off - When this is configured on the egress IB endpoint interface, the B-Tag DEI value will always be "0".

NOTE

There is no force option for the DEI setting.

NOTE
If the packet ingress is through a C-Tagged endpoint destined to a remote PBB node, the C-Tag CFI value will be used as described above.

NOTE
Internal drop precedence on the ingress is derived based on the original packet's PCP value operated by the PCP decode table and optionally merged with the packet's DEI/CFI bit if the "qos use-dei" is set on the ingress port.

I-Tag PCP Setting

I-Tag PCP is always taken from the PCP value that is received from the ingress. In most cases this is the SVLAN PCP value.

NOTE
If the packet ingress is through a C-Tagged endpoint destined to a remote PBB node, the C-Tag PCP value will be used as the I-Tag PCP value in the PBB header.

NOTE
If the packet ingress is through an untagged endpoint destined to a remote PBB node, the PCP value of "0" will be used as the I-Tag PCP value in the PBB header.

I-Tag I-DEI Setting

Packets ingress through an S-Tagged endpoint destined to a remote PBB node will have the DEI bit preserved from the received packet copied unto the PBB header I-Tag I-DEI field.

NOTE
If the packet ingress is through a C-Tagged endpoint destined to a remote PBB node, the C-Tag CFI value will be used as the I-Tag I-DEI value in the PBB header.

NOTE
If the packet ingress is through an untagged endpoint destined to a remote PBB node, the I-Tag I-DEI value in the PBB header will be "0".

S-Tag PCP/DEI Setting

When a packet is received from an IB endpoint, there are PCP/DEI in the B-Tag, PCP/DEI in the I-Tag and optionally PCP/DEI in the S-Tag if S-VLAN keep mode is configured. When this packet is switched to a local S-Tagged endpoint, the PCP/DEI setting will be set according to the rules documented in the following sections.

S-Tag PCP Setting

The packet internal priority for an ingress IB endpoint is derived based on the ingress PCP decode table.

1. S-VLAN keep mode is set - The PBB packet's original S-VLAN's PCP value.
2. S-VLAN Keep mode is not set - The original packet's I-Tag PCP value.

   Depending on the encode PCP policy configured on the egress S-Tagged endpoint, the PCP value will be derived as following:

S-Tag DEI Setting

The packet internal drop precedence for an ingress IB endpoint is derived based on ingress PCP decode table operated on:

1. S-VLAN keep mode is set - The PBB packet's original S-VLAN's PCP value and merged with S-VLAN's DEI bit if "qos use-dei" is also configured at the ingress IB endpoint.
2. S-VLAN Keep mode is not set - The original packet's I-Tag PCP value and merged with I-Tag's I-DEI bit if "qos use-dei" is also configured at the ingress IB endpoint.

   Depending on the encode PCP policy configured on the egress S-Tagged endpoint, the DEI value will be derived as following:

Configuring PBB

This section discusses the limitations and configuration commands required to configure PBB.

Limitations

- Disabling PBB - When PBB is configured, and you want to disable PBB, you will have to remove all endpoint and b-dest-mac configurations. The alternative is to delete the VPLS instance before disabling PBB on the instance.
- Ensure the BVLAN configured for the vlan-isid endpoint is already configured as part of the Layer 2 VLAN for the given interface.
- If PBB is not configured before the endpoint is configured, the user will need to delete all endpoints before configuring PBB.
- A dual tagged endpoint configuration is not allowed under the PBB configuration.
- Auto-discovery configuration is not allowed under the PBB configuration.
- Multicast configuration is not allowed under the PBB configuration.
- VPLS-MTU configuration is not allowed under the PBB configuration.
- VPLS peer configuration is not allowed under the PBB configuration.
- VPLS-local-switching is turned on when PBB is configured. Since PBB support is internally implemented using VPLS local switching, the local switching feature cannot be turned off when PBB is configured for the given VPLS instance.
- VC-mode configuration is not allowed under the PBB configuration.
- ISIDs cannot be reused across VLANs under the PBB configuration.

Configuring PBB

Note that the configuration of PBB on the Brocade NetIron MLX Series and Brocade NetIron XMR Series is done under the VPLS CLI constructs, since PBB on the Brocade NetIron MLX Series and Brocade NetIron XMR Series is internally supported similar to Local VPLS. However, PBB on the Brocade NetIron MLX Series and Brocade NetIron XMR Series does not actually use MPLS. PBB on the Brocade NetIron MLX Series and Brocade NetIron XMR Series does not generate any MPLS signaling and does not use any MPLS protocols. All PBB traffic conforms to the PBB standard.

PBB configuration is not enabled by default and must be enabled per VPLS instance. The optional Backbone Destination MAC Address configuration under PBB is used for PBB connections, which do not require flooding in the PBB core. The customer may use this capability to set the default B-DA to the address of the destination BEB of the point-to-point connection.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)# b-dest-mac 2010.0345.4232
vlan 10
tagged ethe 4/1
vlan 200 isid 20000
tagged ethe 2/1
```

Enabling a new PBB configuration

The following command enables a PBB instance.

```
device(config)#router mpls
device(config-mpls)#vpls vinst 2000
device(config-mpls-vpls-vinst)# pbb
```

Syntax: [no] pbb

PBB Backbone Destination MAC Address Configuration

Configuring a Backbone destination MAC address for a PBB instance allows the flood traffic to be directed towards the specified BEB node instead of using the default backbone multicast destination MAC address. This is used to create point-to-point connections (using a Unicast MAC address).

```
device(config)#router mpls
device(config-mpls)#vpls vinst 2000
device(config-mpls-vpls-vinst)# pbb
device(config-mpls-vpls-vinst-pbb)#b-dest-mac 0001.0001.0003
```

Syntax: [no] b-dest-mac *Ethernet MAC address used for point topoint*

A multicast MAC address will be allowed only if it has the well-known PBB MAC prefix of 01-1E-83.

The following MAC addresses are not allowed:

- A broadcast MAC address is disallowed.
- A zero MAC address is disallowed.
- The own chassis mac in disallowed.

Configuring PBB policy

Any global PBB specific configuration is configured using this sub-mode.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)#
```

Syntax: [no] pbb

System-wide SVLAN Tag Type

The **stag-type** command specifies the Etype used for S-tagged packets in SVLAN-keep mode. This Etype is global.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)#
device(config-mpls-vpls-policy-pbb)#stag-type 8888
```

Syntax: [no] stag-type *hex*

The default Stag-type is 0x88A8.

SVLAN Keep Mode Configuration

To configure SVLAN keep mode, use the **svlan-keep** command. Once the SVLAN keep mode is configured, the system configured SVLAN Etype value will be used to enforce that the SVLAN is present in the IB packets received.

```
device(config)#router mpls
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)#
device(config-mpls-vpls-policy)#pbb
device(config-mpls-vpls-policy-pbb)#
device(config-mpls-vpls-policy-pbb)#svlan-keep
```

Syntax: [no] svlan-keep

```
Before configuring the SVLAN keep mode, ensure that the system-wide SVLAN tag type is already set to the
desired value. If there is a mismatch, the packet will be discarded.
```

Vlan force-pbb-pcp option

The **force-pbb-pcp** command configures the PCP value per VLAN. This command applies only to IB tagged endpoints, whereas the VLAN priority applies to Layer 2 VLANs.

```
device(XMR12)#conf t
device(XMR12)(config)#vlan 60
device(XMR12)(config-vlan-60)#force-pbb-pcp 5
```

Syntax: [no] force-pbb-pcp *value*

The *value* variable set the PCP value per VLAN and can be configured in the range from 1 to 7.

Show Commands

The following section discusses available show commands.

Show mpls vpls detail

The **show mpls vpls detail** command displays information about the operation state of the VPLS instance in regard to the local endpoints.

```
device(XMR12)#show mpls vpls detail
VPLS 1, Id 1, Max mac entries: 8192
 PBB
 Total vlans: 2, Tagged ports: 2 (2 Up), Untagged ports 0 (0 Up)
 IFL-ID: 4096
  Vlan 300
   Tagged: ethe 1/5
  Vlan 3000 isid 40000
   Tagged: ethe 4/1
CPU-Protection: ON,  MVID: 0x001,  VPLS FID: 0x0000a00c
 Local Switching: Enabled
 Extended Counter: ON
```

Show mac

The **show mac** command displays the MACs learned by Layer 2.

```
device(XMR12)#show mac
Total active entries from all ports = 1
Type Code - ST:Static SEC:Secure 1x:Dot1x NA: NotAvail A:Allow D:Deny IB: Installed B-MAC
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
MAC Address     Port      Age      VLAN    Type
0012.f2f7.3b00  4/19         0         10           IB
```

Syntax: show mac

Show mac vpls

The **show mac vpls** command displays the MACs learnt under all VPLS instances.

```
device(XMR12)#show mac vpls
Total VPLS mac entries in the table: 2 (Local: 2, Remote: 0)
VPLS       MAC Address    L/R/IB Port  Vlan(In-Tag)/Peer ISID     Age
====       ===========    ====== ====  ================= ====     ===
1          0000.0404.0000 IB     1/2   300               30000    0
1          0000.0403.0000 L      1/1   200               NA       0
```

Show mac vpls X HHHH.HHHH.HHHH

To view the detail of the B-MAC info, you must enter the command **show mac vpls x** , where *hhhh.hhhh.hhhh* is the C-MAC that has a B-MAC association.

```
device(XMR12)#show mac vpls 1 0000.0404.0000
    VPLS: 1         MAC:  0000.0404.0000     Age:     0
    Local MAC       Port: 1/2                VLAN:    300 ISID:    30000
    Associated B-MAC: 0000.B000.0201
    Trunk slot mask: 0x00000000
```

Syntax: show mac vpls x *hhhh.hhhh.hhhh*

Show mac vpls X b-mac HHHH.HHHH.HHHH

The **show mac vpls X b-mac** command is used to display the C-MACs learned per B-MAC per VPLS instance.

```
device(XMR12)#show mac vpls 1 b-mac 0000.B000.0201
Total VPLS mac entries associated with b-mac 0000.B0000.0201: 2
MAC Address    Port  Vlan(In-Tag)/Peer ISID      Age
===========    ====  ================= ====      ===
0000.0404.0000 1/1   300               30000     10
0000.0408.0000 1/1   200               30000     20
```

Syntax: show mac vpls X b-mac *hhhh.hhhh.hhhh*

hhhh.hhhh.hhhh is the C-MAC that has a B-MAC association.

Show mac vpls pbb-ib x

The **show mac vpls pbb-ib** command displays the C-MACs associated with any B-MAC. Otherwise, the output refers to a specific instance. Here is a sample output:

```
device(XMR12)#show mac vpls pbb-ib 1
Total VPLS mac entries associated with instance 1 pbb-ib: 3
MAC Address     BMAC Address    Port  Vlan(In-Tag)/Peer ISID      Age
============    ============    ====  ================= ====      ===
0000.0404.0000  0000.B000.0201  1/2   300               30000     10
0000.0414.0000  0000.B000.0201  1/2   300               30000     12
0000.0418.0000  0000.B418.0302  1/1   300               30000     20
device(XMR12)#show mac vpls pbb-ib
Total VPLS mac entries: 5
VPLS ID MAC Address     BMAC Address    Port  Vlan(In-Tag)/Peer ISID      Age
======= ============    ============    ====  ================= ====      ===
    3     0000.0000.0200  0000.B020.A101  4/4   120               33000                          30
    4      0000.0003.0220  0000.B004.B001  4/1
121                                                                      2000
22
    1      0000.0404.0000  0000.B000.0201  1/2   300               30000                      10
    1      0000.0414.0000  0000.B000.0201  1/2
300                                                                      30000                      12
    1      0000.0418.0000  0000.B418.0302  1/1   300               30000     20
```

Syntax: show mac vpls pbb-ib *x*

x refers to a VPLS PBB instance. If no instance is specified, the MACs for all instances are displayed, sorted by CMAC.

Show nht

The **show nht** command will show those entries that are MAC based. For MAC based entry many of the fields displayed are not applicable and will have the N.A. value displayed.

```
device(XMR12)#show nht
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP          Index      MAC Address                    VLAN      Out I/F Out Port TNL CNT XC CNT LABEL
EXP/PCP
10.20.20.1          0               000c.dbf5.c773    1           1/20           1/20
0                        1
N.A.              1           000c.dbf4.0000    50   N.A.      N.A.
0                         3                                  0
N.A.              2           000c.dbf5.0000    50   N.A.      N.A.
0                         2                                  0
N.A.              3           000c.dbf6.0000    60   N.A.      N.A.
0                         1                                  5
```

Syntax: show nht

Show nht vlan vlan_id

The **show nht vlan** command allows the nht entry to be displayed based with those that have the matching VLAN ID.

```
device(XMR12)#show nht vlan 50
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP Index MAC Address              VLAN Out I/F  Out Port  TNL CNT XC CNT  LABEL   EXP/PCP
N.A.            1 000c.dbf4.0000 50   N.A.     N.A.      0       3               0
N.A.            2 000c.dbf5.0000 50   N.A.     N.A.      0       2               0
```

Syntax: show nht vlan *vlan_id*

Show nht mac-based [vlan vlan id]

The **show nht mac-based** command displays the MAC-based NHT entries as well as optionally filtered to a specified VLAN ID.

```
device(XMR12)#show nht mac-based
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP Index     MAC Address          VLAN Out I/F  Out Port  TNL CNT XC CNT  LABEL   EXP/PCP
N.A.        1      000c.dbf4.0000 50   N.A.     N.A.      0       3               0
N.A.        2      000c.dbf5.0000 50   N.A.     N.A.      0       2               0
N.A.        3      000c.dbf6.0000 60   N.A.     N.A.      0       1               5
device(XMR12)#show nht mac-based vlan 60
Reconcile Done -
  ARP = 0, GRE = 0, MPLS = 0, phase_1 = 0, l2vpn = 0, phase_2 = 0
NHT IP Index     MAC Address          VLAN Out I/F  Out Port  TNL CNT XC CNT  LABEL   EXP/PCP
N.A.        3      000c.dbf6.0000 60   N.A.     N.A.      0       1               5
```

Syntax: show nht mac-based [**vlan** *vlanid*]

Show cam ifl-isid slot/port output changes

The output of the **show cam ifl-isid** command displays the Service-Type programmed in the service PRAM.

```
device(XMR12)#show cam ifl-isid 1/2
Slot Index   Port  Outer VLAN Itag ISID  PRAM   IFL ID IPV4/V6  Service
     (Hex)                              (Hex)          Routing  Type
1    00c1ffb 1/2   300        2000       181ffb 4097   0/0      5
1    00c1ffc 1/1   300        2000       181ffc 4097   0/0      5
```

Syntax: show cam ifl-isid *slot/port*

Show service-type-table output changes

The output of **show service-type-table** command displays the Forced VLAN Action field programmed in the service type table entry.

```
device#show service-type-table port 1/1
ST   Service   Service  VLAN   RX QOS  TCI    TC    VLAN  IPv4   IPv6   Mcast Forced
entry   type                 ID    valid    ID   valid  Index   ID    Rtg    Rtg   bidir  VL ACT
0x00000  0 (LEGACY) 000000    1     00000   0      -    0000    N      N     N      N
0x00032  1 (VPLS  ) 000005    1     04095   0      -    0050    Y      Y     N      N
0x000c8  1 (VPLS  ) 000003    1     04095   0      -    0200    Y      Y     N      Y
```

Syntax: show service-type-table *slot/port*

802.1ag over PBB OAM

Connectivity Fault Management (802.1ag) is for end-to-end connectivity monitoring. The following functionality has been added.

- CFM monitoring for C-SVLAN/S-VLAN
- CFM monitoring for ISID and B-VLAN
- CFM monitoring for Link MA
- Port status TLV
- Remote Defect Indication

NOTE
PBB-OAM is not supported in the 24x10G card. Only Plain VLAN CFM is supported in the 24x10G card.

Configuration scenarios

Within a PBBN (see MAC Learning for PBB Packets on page 302), the encapsulation performed by Provider Instance Ports (PIPs) also encapsulates CFM frames sourced by customers attached to Customer Network Ports (CNPs). The encapsulation of S-VLAN and C-VLAN CFM frames hides them from the PBBN. All eight levels of CFM frames generated in customer networks are carried over the backbone as encapsulated data and may be used by customer networks.

FIGURE 56 Backbone Edge Bridge Operation

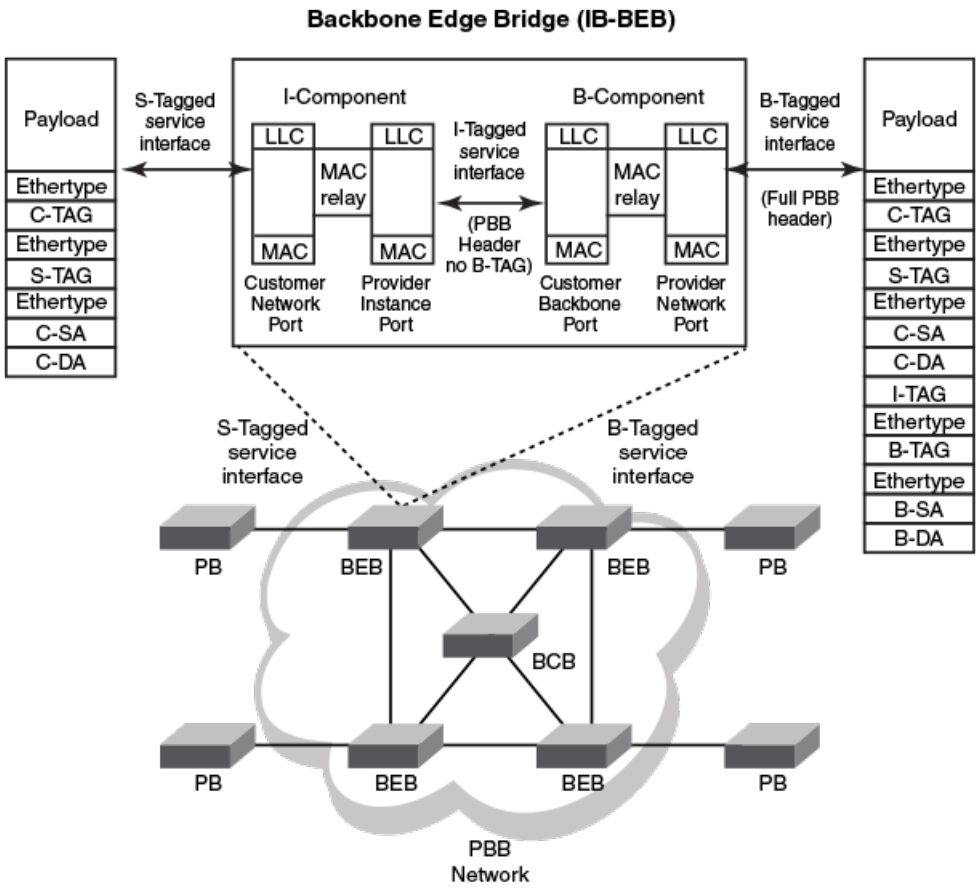
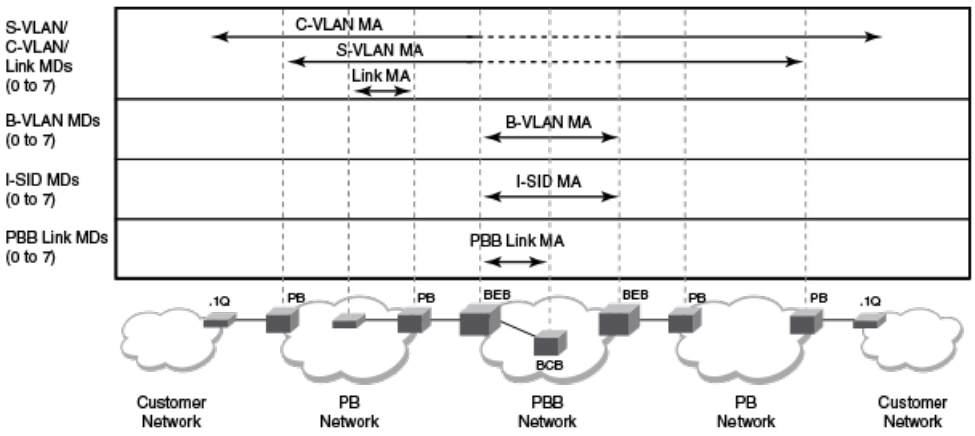


Figure 59 describes the four types of Maintenance Associations (MA) defined by IEEE 802.1ag CFM.

- A full set of eight MD levels exists within each PBBN for use by I-SIDs MAs.
- A full set of eight maintenance levels exists within each PBBN for use by B-VLAN CFM frames.
- An additional eight maintenance levels exists for the LAN link segments.

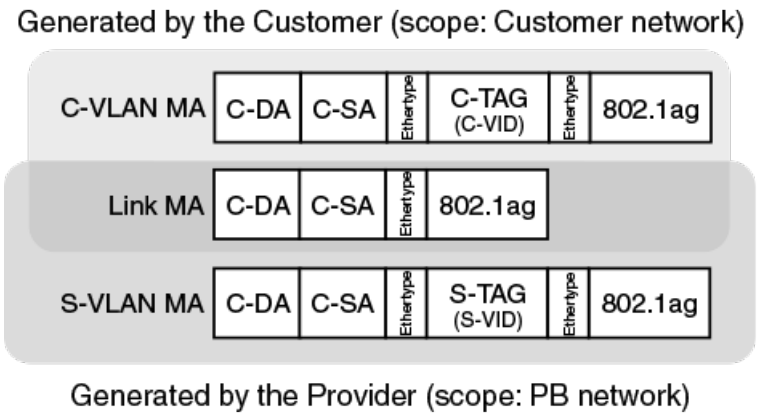
FIGURE 57 Maintenance Association (MA) Categories



S-VLAN, C-VLAN, and Link MAs (Figure 60)

- Customers and Provider (PB) share the same MD level space.
- CFM processing scope within Customer/PB network is determined by the MD level selection.
- The PB network normally only processes provider MD level OAM frames.
- Customer C-VLAN OAM frames use customer MD level and are normally not processed by the PB network.
- S-VLAN CFM frames are only visible where S-TAGs are processed.
- C-VLAN CFM frames are only visible where C-TAGs are processed.
- Once the S-VLAN CFM frames are encapsulated by a BEB, they appear just like any data frame within the PBBN. That is, they do not activate any CFM functions within the PBBN past the BEB.

FIGURE 58 S-VLAN, C-VLAN, and Link MA Frames



B-VLAN MAs (Figure 61)

- B-VLAN MAs manage the B-VLANs within a single PBBN.

- The scope of these 802.1ag frames is the PBB network only. These frames do not leave the PBB network.
- These CFM frames are only visible within the PBBN where the B-TAG is being processed.

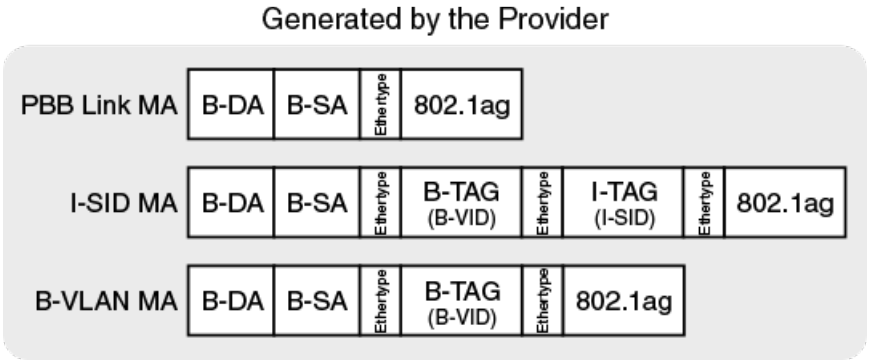
I-SID MAs (Figure 61)

- Backbone service instance CFM frames are only visible within the PBBN where the I-TAG is being processed.

PBB LAN Link Segment MAs (Figure 61)

- PBB LAN link MAs optionally manage links within a PBBN.

FIGURE 59 PBB MA Frames



Types of MEPs and MIPs

Associated with the MAs described in the previous section, there are MEPs and MIPs. Support is provided for the following types of MIPs and MEPs on the appropriate interfaces:

- MIPs and up/down MEPs for C-VIDs: Where C-Tags are processed
- MIPs and up/down MEPs for S-VIDs: Where S-Tags are processed
- Up/down MEPs for I-SIDs: Where I-Tags are processed
- MIPs and up/down MEPs for B-VIDs: Where B-Tags are processed
- Down MEPs for Link MAs

Hierarchical Fault Detection Operation

Ethernet OAM uses a hierarchical fault detection scheme.

- Customer scope: Uses C-VLAN MA and Link MA
- PB scope: Uses S-VLAN MA and Link MA
- PBB scope: Uses B-VLAN MA, I-SID MA, and Link MA

Faults can be detected using Continuity Check messages at any of the four MA categories. Once a fault is detected, Link Trace can be used to narrow down the location of the fault. Depending on the location of the fault, a different MA category may need to be used to further isolate the location of the fault.

802.1ag for Link MA

A new MA type is supported for Link MA.

```
device(config- dot1ag -DONAME) #ma-name MANAME link-MA priority 4
device(config- dot1ag -DONAME-MANAME) # mep 2 down port eth 1/1
```

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | [**vll** [*vll-name*] | [*vll-id*] | **vll-local** [*vll-name*] | **link-MA** } **priority** *priority*

Only the **down MEP** configuration is allowed for a link-MA.

Syntax: mep *mep-id* [**up** | **down**] [**vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

Error messages

If an **up MEP** command is issued the following error message is displayed.

```
device(config- dot1ag -DONAME-MANAME) # mep 2 up port eth 1/1
Error : UP MEP cannot be configured on LINK-MA
```

MEP configuration for link MA is rejected if more than one port is entered in the configuration.

```
device(config- dot1ag -DONAME-MANAME) # mep 2 down port eth 1/1 to 1/3
Error : MEP cannot be configured on multiple ports for Link MA
```

MIP operation is not allowed for link MA. An error is displayed if the **MIP-POLICY** command is issued.

```
device(config- dot1ag -DONAME-MANAME)#mip-policy explicit
Error: MIP cannot be configured on LINK-MA
```

Link MA capabilities and limitations

- MIP functionality is not supported for link-MA.
- Link trace functionality is not supported for link-MA.
- Loopback and delay measurement functionality is supported on link-MA.
- 802.1ag sub-second-timer functionality is supported for link-MA.
- 802.1ag functionality is supported during hitless upgrade and switchover.
- 802.1ag functionality is supported for LAG.
- When MEP is configured on a VPLS instance with sub-second CCM timer, the MIP flooding should be turned off. When there is no MEP and it is a sub-second-timer, then dot1ag flooding should be turned on.

```
device#show cfm
Domain: D10
Index: 3
Level: 7
  Maintenance association: MA
  Ma Index: 1
  CCM interval: 10000 ms
  LINK MA ID: 0
  Priority: 3
  MEP                 Direction                    MAC                              PORT          PORT-
STATUS-TLV
  ====              ========                 =========                       =====
===============
    10      DOWN                 0012.f2f7.3900             ethe 1/1        N
device#show cfm connectivity
Domain: D10 Index: 3
Level: 7
  Maintenance association: MA
  MA Index: 1
  CCM interval: 10000 ms
  LINK MA ID: 0
  Priority: 3
```

```
RMEP            MAC                           VLAN/PEER        AGE
PORT                      SLOTS                     STATE
====      ==========                             ===========     =====
=====               =====                       =======
1         0012.f2f7.3861                              0                   54020
1/1              1           OK
```

802.1ag for CVLAN and SVLAN

Brocade supports MEP and MIP for the regular VLAN and MEP for VPLS VLAN.

You will need to create a new MA for PBB-VPLS. SVLAN functionality is supported in a similar manner as it is with CVLAN functionality. The only difference is the tag-type configuration for the VPLS end-points.

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | **pbb-vpls** *vpls-id* | [**vll** [*vll-name*] | [vll-id]]**vll-local**[*vll-name*]]**priority***priority*

Example

```
device(config- dot1ag -DONAME)# ma-name MANAME pbb-vpls 10 priority 4
```

Both down MEP and UP MEP configuration are accepted for CVLAN and SVLAN.

Syntax: mep *mep-id* [**up** | **down**] [**vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

- UP MEP for CVLAN and SVLAN will transmit CCM packets on C-tagged, S-tagged and IB-tagged end-points on the service instance.
- MIP will take care of CVLAN/SVLAN translation and MIP is created on C-tagged and S-tagged end-points.
- 802.1ag sub-second-timer functionality is supported for PBB CVLAN/SVLAN.
- 802.1ag functionality for PBB CVLAN/SVLAN is supported during switch-over.
- 802.1ag functionality is not supported during hitless upgrade for PBB CVLAN/SVLAN.
- 802.1ag functionality is supported if the PBB CVLAN/SVLAN end-point is a LAG.
- Link-trace, loopback and delay-measurement for PBB CVLAN/SVLAN is supported.
- 802.1ag for CVLAN/SVLAN is supported on untagged end-points, port based untagged end-points and C-tagged end-points.
- 802.1ag for CVLAN/SVLAN is not supported on dual-tagged end-points.
- SVLAN keep-mode is supported.
- When MEP is configured on a VPLS instance with sub-second CCM timer, the MIP flooding should be turned off. When there is no MEP and it is a sub-second-timer, then dot1ag flooding should be turned on.

```
device# show cfm
Domain: D1
Index: 1
Level: 3
  Maintenance association: MA
  Ma Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  MEP   Direction                    MAC                              PORT       PORT-STATUS-TLV
  ====  ========    =========                                 =====       ==============
    1          UP          0012.f2f7.3900           ethe 1/1                    N
  MIP      VLAN/VC                           Port                          Level                      MAC
  ====    ========                              =====                        =====
==========
              200                   1/1               3               0012.f2f7.3901
device# show cfm connectivity
Domain: D1 Index: 1
Level: 3
  Maintenance association: MA
```

```
  MA Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  RMEP              MAC                   VLAN/ISID                  AGE
PORT              SLOTS           STATE
  ====              =====                    =======                   ======
=======               ====              =========
    10          0012.f2f7.3860          200                           73460
1/2               1           OK
```

802.1ag for BVLAN

BVLAN functionality is supported by the use of regular VLANs only.

802.1ag for ISID

PBB-VPLS functionality supports IB-tagged end-points. 802.1ag for ISID is supported in a similar manner as it is supported for CVLAN and SVLAN end-points for local-VPLS. It requires the creation of a new MA for PBB-VPLS.

```
device(config- dot1ag -DONAME)# ma-name MANAME pbb-vpls 10 priority 4
```

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | **pbb-vpls** *vpls-id* | [**vll** [*vll-name*] | [*vll-id*] | **vll-local** [*vll-name*] } **priority** *priority* **mep** *mep-id* [**up** | **down**] [**vlan** *vlan-id* **ISID** *isid* **port** *port-id* | **vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

- Both down MEP and UP MEP configuration is accepted for ISID.
- UP MEP for ISID will transmit CCM packets out ONLY on the BVLAN ports carrying the same ISID.
- MIP functionality is not supported for ISID.
- 802.1ag sub-second-timer functionality for ISID is supported.
- 802.1ag functionality for ISID during switchover is supported.
- 802.1ag functionality for ISID during hitless upgrade is not supported.
- 802.1ag functionality for ISID is supported where IB-tagged end-points are LAG.
- Link-trace, loopback and delay-measurement for ISID is supported.
- SVLAN keep-mode is supported.
- When MEP is configured on a VPLS instance with sub-second CCM timer, the MIP flooding should be turned off. When there is no MEP and it is a sub-second-timer, then dot1ag flooding should be turned on.

```
device#show cfm
Domain: D2
Index: 2
Level: 1
  Maintenance association: MA
  Ma Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  MEP           Direction                       MAC                    PORT                      PORT-
STATUS-TLV
  ====      ========                        =========                      ======
==============
    10          DOWN              0012.f2f7.3901               ethe
1/2                                   N

  MIP     VLAN/VC                   Port                    Level                     MAC
  ====    ========                =====                 =====                  ==========
               100                1/1                1              0012.f2f7.3901
               200                1/2                1              0012.f2f7.3901
```

In the following example, MIP is created for BVLAN not for ISID.

```
device# show cfm connectivity
Domain: D2 Index: 2
Level: 1
  Maintenance association: MA
  MA Index: 1
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 3
  RMEP           MAC                        VLAN/ISID                    AGE
PORT               SLOTS               STATE
  ====  ==========                          =========                        =====
=====               ======                  ======
     1      0012.f2f7.3861                        1000                          72320
1/2               1               OK
```

802.1ag Port Status TLV

The Port Status TLV indicates the ability of the Bridge Port on which the transmitting MEP resides to pass ordinary data, regardless of the status of the MAC.

> **NOTE**
> Port Status TLV is not be supported for LINK MA.

MEP configuration provides support for port status TLV.

Syntax: mep *mep-id* [**up** | **down**] [**tlv-type** *tlv-type-value*] [**vlan** *vlan-id* **ISID** *isid* **port** *port-id* | **vlan** *vlan-id* **port** *port-id* | **port** *port-id*]

The *tlv-type-value* should be set to the *port-status-tlv*.

Sample configuration output

Syntax: show cfm domain *domainName* **ma** *maName* **mep-id** *mepId*

```
Brocade # show cfm domain D2 ma 1 mep-id 5000
Domain: customer
Index: 1
Level: 7
  Maintenance association: admin
  Ma Index: 2
  CCM interval: 10000 ms
  ESI aaa VLAN ID: 100
  Priority: 7
  MEP   Direction           MAC                                          PORT         PORT-STATUS-TLV
  ====  =========  =========                                          =======        ================
    1      DOWN               0024.3863.7741           ethe 1/1            Y
Y - Means port status tlv is enabled for the MEP
N - Means port status tlv is not enabled for the MEP
Brocade #show cfm connectivity do bvlan ma bvlan rmep-id 5000
Domain: bvlan Level: 5
Maintenance association: bvlan VLAN VLAN/VPLS/VLL ID: 250 Priority: 5
CCM interval: 1000 ms
RMEP           MAC                 PORT      Oper      Age          CCM              RDI
Port      Intf      Intvl Seq          core
                           State     Val           Cnt      Status Status      Error Error Fault level
====  ============== ==== =====  ==== ====== =======  ==== ====== ====== === ==
5000  0024.3898.da20           3/1      OK               2156 216030      N
2      0     N     N           N
Value              Port Status

1              Port Blocked ( Not Forwarding)

2              Port Forwarding
```

802.1ag RDI

The Remote Defect Indication (RDI), a single bit, is carried in CCM packets. The absence of RDI in a CCM indicates that the transmitting MEP is receiving CCMs from all RMEPs. The presence of RDI indicates that transmitting MEP is not receiving CCM from one or more RMEPs.

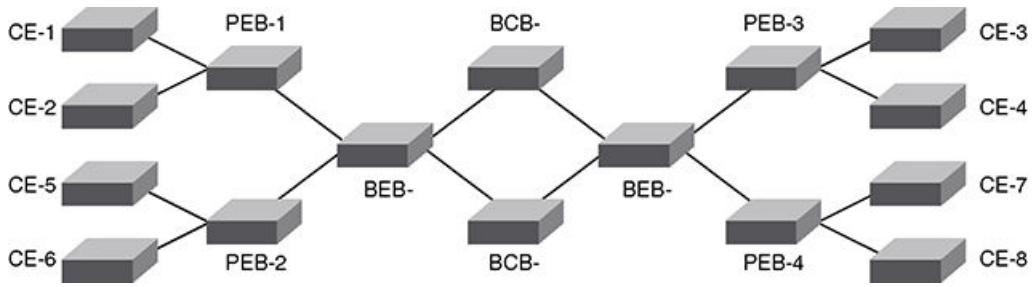
```
device#show cfm connectivity domain customer ma admin rmep-id 1
Output:
Domain: customer Level: 7
Maintenance association: admin VLAN VLAN/VPLS/VLL ID: 100 Priority: 7
CCM interval: 100ms
                              PORT               Oper          Age
RMEP         MAC                        State         Val                Cnt                CCM
RDI
====          =========                  ======                 =====      =====          ======= ==========
  1          0024.3863.7745      1/1                              OK           8180
13               Y
Port                Intf               Intvl           Seq
Status                Status               Error           Error
======                ======               =====           =====
0                0               N           N

Sample Output:
CES-2#show cfm connectivity domain customer ma admin rmep-id 1
Output:
Domain: customer Level: 7
Maintenance association: admin VLAN VLAN/VPLS/VLL ID: 100 Priority: 7
CCM interval: 100ms
RMEP   MAC              PORT    Oper     Age       CCM           RDI
                                                   State    Val
Cnt
====  =========   ====== =====   =====  ======= ==========
  1  0024.3863.7745    1/1    OK          8180         13              Y
Port        Intf         Intvl     Seq
Status    Status     Error    Error
====== ======   =====  =====
   0                0          N         N
```

Deployment Scenarios and CLI Configuration

NOTE
RDI is not supported for LINK MA.

Deployment Scenario 1 (Down MEPs on CEs and MIP on PE)

FIGURE 60 Deployment scenario 1



Configuration for CE Devices

Configuration for CE devices is the same as for 802.1ag over VLAN.

Configuring CE-1

VLAN configuration

```
device(config)#vlan 10
device(config-vlan-10)#tagged ethernet 1/1
```

CFM configuration:

1. To enable CFM, enter the following command:

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   device(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain of vlan-id 10 with a priority 3.

   ```
   device(config-cfm-md-CUST_1)#ma-name ma_5 vlan-id 10 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages. The default is 10-seconds.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

5. Configure a MEP on port 1/1 and vlan 10.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#mep 1 down vlan 10 port ethe 1/1
   ```

6. Configure a remote-mep.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#remote-mep 2 to 2
   ```

7. Continue the configuration for other CE devices as needed.

Configuration for PE Devices:

Configuring PEB-1

Tag-type configuration

Assume CVLAN tag-type is 0x8100 and S-VLAN tag-type is 0x900.

```
device(config)#tag-type 8100 eth 1/1
device(config)#tag-type 8100 eth 1/2
device(config)#tag-type 9100 eth 1/3
```

VPLS local configuration:

Assume CE-1 is connected to PEB-1 port 1/1 on VLAN 10 and CE-2 is connected to PEB-2 port 1/2 on VLAN 20 and PEB-1 is connected to BEB-1 on port 1/3 VLAN 30.

```
device(config)#router mpls
device(config-mpls)#vpls PB-VPLS 20
device(config-mpls-vpls-PB-VPLS)#pbb
device(config-mpls-vpls-PB-VPLS-pbb)#exit
device(config-mpls-vpls-PB-VPLS)#
device(config-mpls-vpls-PB-VPLS)#vlan 10
device(config-mpls-vpls-PB-VPLS-vlan-10)#tagged eth 1/1
device(config-mpls-vpls-PB-VPLS-vlan-10)#
```

```
device(config-mpls-vpls-PB-VPLS-vlan-10)#vlan 20
device(config-mpls-vpls-PB-VPLS-vlan-20)#tagged eth 1/2
device(config-mpls-vpls-PB-VPLS-vlan-20)#
device(config-mpls-vpls-PB-VPLS-vlan-20)#vlan 30
device(config-mpls-vpls-PB-VPLS-vlan-30)#tagged eth 1/3
```

CFM configuration:

If the VPLS local configuration is not done prior to configuring maintenance association. The MA configuration is not allowed. PEB-1 will work as a MIP.

Syntax: ma-name MANAME { **vlan-id** *vlan-id* | **vpls-id** *vpls-id* | **pbb-vpls** *vpls-id* | [**vll** [*vll-name*] | [*vll-id*] | **vll-local** [*vll-name*] } **priority** *priority*

1. Enable CFM using the **cfm-enable** command.

   ```
   device(config)#cfm-enable
   ```

2. Create a maintenance domain with a specified name CUST_1 and level 7.

   ```
   device(config-cfm)#domain-name CUST_1 level 7
   ```

3. Create a maintenance association within a specified domain for vpls-id 20 with a priority 3.

   ```
   device(config-cfm-md-CUST_1)#ma-name ma_5 pbb-vpls 20 priority 3
   ```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10 seconds.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)#ccm-interval 10-second
   ```

   In the above configuration, MIP gets created by default on the VPLS end-points. You can also configure **explicit-mip** on PEB-1. In that case, MIP will be created on the VPLS end-points if a MEP is created on the port at some lower MD Level.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)# mip-creation explicit
   ```

   To change back to default use the following command.

   ```
   device(config-cfm-md-CUST_1-ma-ma_5)# mip-creation default
   ```

Verifying Connectivity Using 802.1ag

Once you configure CE-1, CE-2, PEB-1, and PEB-2 you can determine the end-to-end connectivity by looking at the remote-mep status using the following show commands:

Syntax: show cfm connectivity [**domain** NAME] [**ma** MA NAME]

```
device#show cfm connectivity
Domain: CUST_1 Level: 7
Maintenance association: ma_5
CCM interval: 10000 ms
VLAN ID: 30
Priority: 3
RMEP              MAC                 VLAN/PEER           AGE              PORT           SLOTS
====          ==============          =======                      ====   ====           ====
2             000c.dbe2.8a00                      30             879          1/2              1
device#show cfm connectivity domain CUST_1 ma ma_5 rmep-id 2
Domain: CUST_1 Level: 7
Maintenance association: ma_5 VLAN ID: 30 Priority: 3
CCM interval: 10
RMEP            MAC                 PORT           Oper          Age      CCM            RDI
Port               Intf            Intvl Seq
                                        State           Val          Cnt      Status Status
Error            Error
==== =============== === === ===== ===  ===            ====            ======      ======           ====
```

```
2     000c.dbe2.8a00                1/1 OK  26000 2600 N
0                                    0      N                          N
Verifying Connectivity Using 802.1ag Loopback/Linktrace
```

Use the **cfm linktrace domain NAME ma MA-NAME src- mep** and the **cfm loopback domain NAME ma MA-NAME scr-mep** commands to manually monitor the status of peer, as shown below:

Syntax: cfm linktrace domain NAME ma MA-NAME src- mep *mep-id* **target-mip HH:HH:HH:HH:HH:HH** | **target-mep** *mep-id* } [**timeout** *timeout*] [**ttl** *TTL*]

```
device#cfm linktrace domain CUST_1 ma ma_5 src-mep 1 target-mep 2
Linktrace to 000c.dbe2.8a00 on Domain CUST_1, level 4: timeout 10ms, 8 hops
------------------------------------------------------------------
Hops         MAC          Ingress  Ingress Action  Relay Action
Forwarded  Egress   Egress Action    Nexthop
------------------------------------------------------------------
Type Control-c to abort
1   000c.dbe2.8a00 10.1.1.1   IgrOK          RLY_HIT
Not Forwarded
Destination 000c.dbe2.8a00 reached
```

Syntax: cfm loopback domain NAME ma MA-NAME scr-mep *mep-id* { **target-mip HH:HH:HH:HH:HH:HH** | **targetmep** *mep-id* } [**number** *number*] [**timeout** *timeout*]

```
device#cfm loopback domain CUST_1 ma ma_5 src-mep 1 target-mep 2
DOT1AG: Sending 10 Loopback to 000c.dbe2.8a00, timeout 10000 msec
Type Control-c to abort
Reply from 000c.dbe2.8a00: time=3ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time=38ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
Reply from 000c.dbe2.8a00: time<1ms
A total of 10 loopback replies received.
Success rate is 100 percent (10/10), round-trip min/avg/max=0/4/38 ms.
```

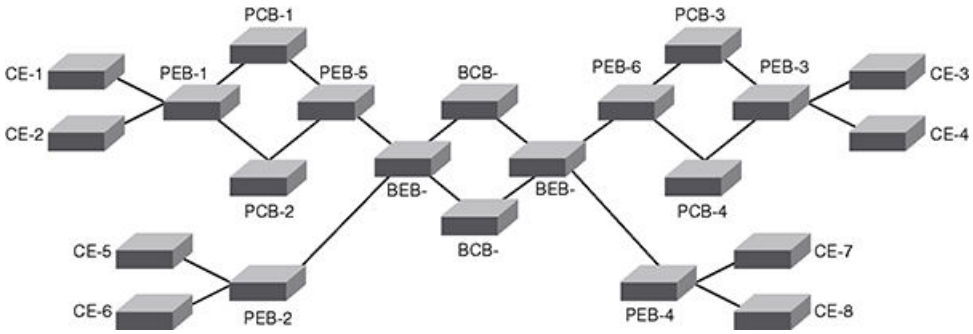
If the linktrace and loopback to target-mep 2 fails, then linktrace can be done on the MIPs on PEB-1 and PEB-2 to know the exact failure.

Deployment Scenario-2 (UP MEPs and MIPs on PEs)

If you have a deployment scenario where PEB-1 is not directly connected to BEB-1, but it is connected to a PB network, then you can use MIPs configured over the intermediary nodes, assuming the PB network is managed by the same administrator.

If the network is managed by a separate administrator, then you can have UP-MEPs configured on the PE ports connected to CE devices. The intermediate devices will have MIPs configured.

FIGURE 61 Deployment scenario-2and3



Configuration for PE Devices

Configuring PEB-1

The local VPLS configuration will be the same as shown in the previous deployment scenario. If the local VPLS configuration is not done prior to configuring maintenance association, the MA configuration is not allowed. Also, the port and vlan in the MEP configuration should exist in local VPLS configuration prior to MEP configuration. Otherwise, it is not allowed. The port in the MEP configuration can be either a tagged or untagged port already present in the local-VPLS configuration.

CFM configuration steps for PEB-1

1. Enter the **cfm-enable** command to enable CFM.

```
device(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
device(config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

```
device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10-seconds.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

5. Configure a MEP on port 1/1 and vlan 10.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#mep 6 up vlan 10 port ethe 1/1
```

A similar configuration will need to be done on PEB-3. MIP should be configured on PEB-5, PEB-6, PCB-1, PCB-2, PCB-3, and PCB-4.

CFM configuration steps for PEB-5.

1. Enter the **cfm-enable** command to enable CFM.

```
device(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
device(config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

```
device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10-seconds.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

To monitor the connectivity between PEB-1 and PEB-3, you can use the **show cfm connectivity** command as mentioned in the previous scenario. Also, you can use either loopback or linktrace on PEB-1 or PEB-3.

Deployment Scenario-3 (MIPs on BEBs)

In Deployment Scenario-2 (UP MEPs and MIPs on PEs) on page 322 BEBs can work as MIP for the UP MEPs configured on PEs in the previous deployment scenario.

Configuring BEB-1

Tag-type configuration

Assume S-VLAN tag-type is 0x9100 and B-VLAN tag-type is 0x88e8

```
device(config)#tag-type 9100 eth 1/1
device(config)#tag-type 9100 eth 1/2
device(config)#tag-type 88e8 eth 1/3
```

SVLAN Configuration

```
device(config)#vlan 10
device(config-vlan-10)#tagged ethernet 1/1
```

SVLAN Configuration

```
device(config)#vlan 20
device(config-vlan-20)#tagged ethernet 1/2
```

BVLAN Configuration

```
device(config)#vlan 100
device(config-vlan-100)#tagged ethernet 1/3
```

VPLS local configuration

Assume port 1/1 and 1/2 on BEB are connected to PE devices (S-tagged end-point) and 1/3 is an IB-tagged end-point.

```
device(config)#router mpls
device(config-mpls)#vpls PBB-VPLS 20
device(config-mpls-vpls-PBB-VPLS)#pbb
device(config-mpls-vpls-PBB-VPLS-pbb)#exit
device(config-mpls-vpls-PBB-VPLS)#
device(config-mpls-vpls-PBB-VPLS)#vlan 10
device(config-mpls-vpls-PBB-VPLS-vlan-10)#tagged eth 1/1
device(config-mpls-vpls-PBB-VPLS-vlan-10)#
device(config-mpls-vpls-PBB-VPLS-vlan-10)#vlan 20
device(config-mpls-vpls-PBB-VPLS-vlan-20)#tagged eth 1/2
device(config-mpls-vpls-PBB-VPLS-vlan-20)#
device(config-mpls-vpls-PBB-VPLS-vlan-20)#vlan 100 isid 200000
device(config-mpls-vpls-PBB-VPLS-vlan-100-isid-200000)#tagged eth 1/3
device(config-mpls-vpls-PBB-VPLS-vlan-100-isid-200000)#
```

CFM configuration

If the VPLS local configuration is not done prior to configuring the maintenance association, the MA configuration is not allowed. BEB-1 will work as an MIP.

1. Enter the **cfm-enable** command to enable CFM.

```
device(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PROVIDER_1 and level 4.

```
device(config-cfm)#domain-name PROVIDER_1 level 4
```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

```
device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
```

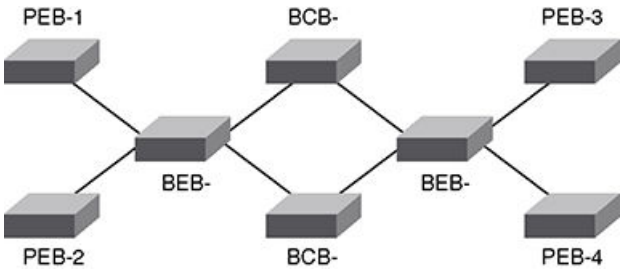
4. Set the time interval between successive Continuity Check Messages. The default is 10-seconds.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

The above configuration will create SVID MIPs on BEBs. You can use linktrace to BEB MIPs from PE MEPs to further isolate the fault location.

Deployment Scenario-4 (ISID MEPs on BEBs)

FIGURE 62 Deployment scenario-4and5



The Local VPLS configuration is similar to the previous scenario.

CFM configuration steps for BEB-1.

1. Enter the **cfm-enable** command to enable CFM

```
device(config)#cfm-enable
```

2. Create a maintenance domain with a specified name PBB_1 and level 4.

```
device(config-cfm)#domain-name PBB_1 level 3
```

3. Create a maintenance association within a specified domain of pbb-vpls 20 with priority 3.

```
device(config-cfm-md-PROVIDER_1)#ma-name ma_8 pbb-vpls 20 priority 3
```

4. Set the time interval between successive Continuity Check Messages (CCM). The default is 10-seconds.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#ccm-interval 10-second
```

5. Configure a MEP on port 1/3 and vlan 10.

```
device(config-cfm-md-PROVIDER_1-ma-ma_8)#mep 6 up ISID 200000 vlan 10 port ethe 1/3
```

To monitor the connectivity between BEB-1 and BEB-2, you can use the **show cfm connectivity** commands as mentioned in the previous scenario. Also, you can use either loopback or linktrace on BEB-1 or BEB-2.

Deployment Scenario-5 (BVLAN MEPs on BEBs and MIP on BCBs)

BVLAN CFM configuration will be similar to regular VLAN and it will support both MEP and MIP functionality.

Show Commands

The **show cfm** command provides the following output.

```
device#show cfm
Domain: md2
Level: 6
  Maintenance association: ma2
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 4
  MEP   Direction       MAC                                          PORT
  ====  =========           =========                                ====
    2    UP                    000c.dbf3.fa02          ethe 1/3
```

The **show cfmconnectivity** command provides the following output.

```
device#show cfm connectivity
Domain: md2 Level: 6
  Maintenance association: ma2
  CCM interval: 10000 ms
  PBB-VPLS ID: 100
  Priority: 4
  RMEP               MAC                     VLAN/PEER                         AGE
PORT              SLOTS
  ====               =====                     ============          =======
========                  ====
    3      000c.dbf3.fb02                           10.2.2.2                   320              1
```

The **show cfm connectivity domain** command provides the following output.

```
device#show cfm co domain md2 ma ma2 rmep 3
Domain: md2 Level: 6
Maintenance association: ma2 PBB-VPLS ID: 100 Priority: 4
CCM interval: 10000 ms
  RMEP   MAC            PORT  Oper     Age      CCM      RDI Port   Intf   Intvl Seq
                               State    Val      Cnt          Status Status Error Error
  ====  ============== ===== =====  ======= ==========  ==== ====== ====== ===== =====
     3  000c.dbf3.fb02  00c35    OK       20         39   N     0      0     N     Y
```

The **show cfm domain** command with the *domain-name***ma***ma-name* parameters, provides the following output.

```
device#show cfm do md2 ma ma2
Domain: md2
Level: 6
  Maintenance association: ma2
  CCM interval: 10000 ms
  VLL ID: 100
  Priority: 4
  MEP   Direction  MAC            PORT/TX PORT
  ====  =========  =========      ====
```

```
     2    UP      000c.dbf3.fa02  ethe 1/3
  REMOTE MEP id 3 MAC 000c.dbf3.fb02    OK 2
CFM port (VL100) PBB-VPLS  100
CFM port (VL800000) PBB-VPLS 100
device#
```

Syntax: show cfm domain *domain-name* ma *ma-name*

Configuring Spanning Tree Protocol

The Brocade NetIron CES Series and Brocade NetIron CER Series devices support the Ethernet Service Instance (ESI) framework. A user can configure ESIs in the process of configuring Provider Bridging and Provider Backbone Bridging. By default, a device has a "default ESI" configured in which VLANs 1- 4090 exist. This chapter refers to configuration and use of Spanning Tree Protocols under the default ESI. For configuration of user-defined ESIs, please refer to the ESI framework, which is described in detail in the *Ethernet Service Instance for Brocade NetIron CES Series and Brocade NetIron CER Series Devices* chapter.

IEEE 802.1D Spanning Tree Protocol (STP)

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

Enabling or disabling STP

STP is disabled by default on the Brocade device. Thus, new VLANs you configure on the Brocade device have STP disabled by default. Table 51 lists the default STP states for the Brocade device.

TABLE 51 Default STP states

Device type	Default STP type	Default STP state	Default STP state of new VLANs
Brocade	Brocade's multiple instances of spanning tree	Disabled	Disabled

By default, each VLAN on a Brocade device runs a separate spanning tree instance. Each Brocade device has one VLAN (VLAN 1) by default that contains all of its ports. However, if you configure additional port-based VLANs on a Brocade device, then each of those VLANs on which STP is enabled and VLAN 1 all run separate spanning trees.

You can enable or disable STP on the following levels:

- **Globally** - Affects all VLANs on the Brocade device.
- **Individual VLAN** - Affects all ports within the specified VLAN. When you enable or disable STP within a VLAN, the setting overrides the global setting. Thus, you can enable STP for the ports within a VLAN even when STP is globally disabled, or disable the ports within a port-based VLAN when STP is globally enabled.
- **Individual port** - Affects only the individual port. However, if you change the STP state of the primary port in a LAG group, the change affects all ports in the LAG group.

Enabling or disabling STP globally

Use the following methods to enable or disable STP on the Brocade device on which you have not configured VLANs.

NOTE
When you configure a VLAN, the VLAN inherits the global STP settings. However, once you begin to define a VLAN, you can no longer configure standard STP parameters globally using the CLI. From that point on, you can configure STP only within individual VLANs.

NOTE
Reloading the Brocade device with global STP enabled can display error on boot-up (error - no more stp instances available) if the number of vlans in the configuration are more than configured **system-max** for STP instances. The error message has no effect on the functionality.

When configuring spanning- tree at the global CLI level, the following message will prompt you to enter "y" for yes or "n" for no to change the spanning-tree behavior at the global level:

```
device(config)#spanning-tree
This will change the spanning-tree behavior at the global level.
Are you sure? (enter 'y' or 'n'): y
```

Enter 'y' to change the spanning-tree behavior. Enter 'n' to make no change to the spanning-tree configuration at the global level.

This command enables a separate spanning tree in each VLAN, including the default VLAN.

Syntax: [**no**] **spanning-tree**

Enabling or disabling STP on a VLAN

Use the following procedure to disable or enable STP on a Brocade device on which you have configured a VLAN. Changing the STP state in a VLAN affects only that VLAN.

To enable STP for all ports in a port-based VLAN, enter commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree
that there is no effect on the functionality due to this error message
```

Syntax: [no] spanning-tree

Enabling or disabling STP on a port

Use the following procedure to disable or enable STP on an individual port.

NOTE
If you change the STP state of the primary port in a LAG group, the change affects all ports in the LAG group.

To enable STP on an individual port, enter commands such as the following.

```
device(config)# interface 1/1
device(config-if-e1000-1/1)# spanning-tree
```

Syntax: [no] spanning-tree

STP in a LAG

The STP standard indicates that by default the path cost is determined by link speed. For a1 G port the path cost is 4 and for 10G port the path cost is 2. However, if a LAG is made consisting of *n* 1G ports, where *n* is less than 10, the path cost remains as 4. The standard does not indicate pathcost explicitly for LAG interfaces or for bandwidths between standard port bandwidth values, (for example, between 1G and 10G). Therefore, during STP deployment you may find that though a LAG has greater bandwidth, its in blocking/ discarding state as its pathcost is the same as any 1G link and the portIndex of 1G port is lower, making the LAG go into a blocking/

discarding state. This behavior is not restricted to 1G or 10G link speed but span across different link speeds. The same behavior also holds TRUE for RSTP deployments.

Default STP bridge and port parameters

Default STP bridge and port parameters lists the default STP bridge parameters. The bridge parameters affect the entire spanning tree. If you are using MSTP, the parameters affect the VLAN. If you are using SSTP, the parameters affect all VLANs that are members of the single spanning tree.

NOTE
STP information is specific to a VLAN, and the Multi-Service IronWare software uses the CONTROL VLAN to get the STP information for the all MIBs under dot1dStp and dot1DStpPortTable. Due to the limitation of the MIBs, information of per STP implementation of every specific VLAN is not displayed.

1. Forward Delay
2. The period of time a bridge will wait (the listen and learn period) before beginning to forward data packets.
3. seconds Possible values: 4 - 30 seconds
4. Maximum Age
5. The interval a bridge will wait for a hello packet from the root bridge before initiating a topology change.
6. seconds Possible values: 6 - 40 seconds
7. Hello Time
8. The interval of time between each configuration BPDU sent by the root bridge.
9. seconds Possible values: 1 - 10 seconds
10. Priority
11. 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.
12. 768 Possible values: 0 - 65535

NOTE
If you plan to change STP bridge timers, it is recommended that you stay within the following ranges, from section 8.10.2 of the IEEE specification: - 2 * (forward_delay -1) >= max_age- max_age >= 2 * (hello_time +1)

Table 52 lists the default STP port parameters. The port parameters affect individual ports and are separately configurable on each port.

TABLE 52 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 1 Gigabit - 4 10 Gigabit - 2

TABLE 52 Default STP port parameters (continued)

Parameter	Description	Default and valid values
		40 Gigabit - 1 100 Gigabit - 1 Possible values are 1- 65535

Changing STP bridge parameters

To change a Brocade device's STP bridge priority to the highest value, so as to make the Brocade device the root bridge, enter the following command.

```
device(config)# vlan 20
device(config-vlan-20)# spanning-tree priority 0
```

To make this change in the default VLAN, enter the following commands.

```
device(config)# vlan 1
device(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 Changing STP bridge parameters.

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.

```
device(config)# vlan 10
device(config-vlan-10)# spanning-tree ethernet 1/5 path-cost 15 priority 64
```

Syntax: [no] spanning-tree ethernet slot/portnum path-cost value | priority value | disable | enable

The **ethernet** *slot/portnum* parameter specifies the interface.

For descriptions of path cost and priority, their default and possible values, refer to Default STP bridge and port parameters on page 331. If you enter a priority value that is not divisible by four, the software rounds it to the nearest value.

The **disable and enable** parameter disables or re-enables STP on the port. The STP state change affects only this VLAN. The port's STP state in other VLANs is not changed.

Root Guard

A new security feature has been added that allows a port to run STP but does not allow the connected device to become the Root. The Root Guard feature provides a way to enforce the root bridge placement in the network and allows STP to interoperate with user network bridges while still maintaining the bridged network topology that the administrator requires. Errors are triggered if any change from the root bridge placement is detected.

NOTE
The feature is also available for MSTP and RSTP.

When Root Guard is enabled on a port, it keeps the port in designated FORWARDING state. If the port receives a superior BPDU, which is a Root Guard violation, it sets the port into BLOCKING state and triggers a Syslog message and an SNMP trap. No further traffic will be forwarded on this port. This allows the bridge to prevent traffic from being forwarded on ports connected to rogue or wrongly configured STP or RSTP bridges.

Root Guard should be configured on all ports where the root bridge should not appear. In this way, the core bridged network can be cut off from the user network by establishing a protective perimeter around it.

Once the port stops receiving superior BPDUs, Root Guard will automatically set the port back to a FORWARDING state after the timeout period has expired.

> NOTE
> Root Guard may prevent network connectivity if improperly configured. It needs to be configured on the perimeter of the network rather than the core. Also, Root Guard should be configured only on the primary port of a LAG. If a port configured with Root Guard is made a secondary port, the LAG deployment will be vetoed.

Enabling Root Guard

Root Guard is configured on a per interfaces basis. To enable Root Guard, enter a command such as the following.

```
device(config)# interface ethernet 5/5
device(config-if-e10000-5/5) spanning-tree root-protect
```

Syntax: [no] spanning-tree root-protect

Enter the **no** form of the command to disable Root Guard on the port.

Setting the Root Guard timeout period

To configure the Root Guard timeout period globally, enter a command such as the following.

```
device(config)# spanning-tree root-protect timeout 120
```

Syntax: [no] spanning-tree root-protect timeout timeout in seconds

The *timeout in seconds* parameter allows you to set the timeout period. The timeout period may be configured to anything between 5 and 600 seconds. Default is 30 seconds.

Checking if Root Guard is configured

To determine if Root Guard is configured, enter the following command.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is enabled
, STP BPDU Guard is disabled
```

Syntax: show interface ethernet slot/port

Displaying the Root Guard state

To display the Root Guard state, enter the **show spanning-tree root-protect** command.

```
device#show spanning-tree root-protect
Port VLAN Current State
13/6 3 Consistent state
13/9 2 Inconsistent state (29 seconds left on timer)
```

Syntax: show spanning-tree root-protect

Reconfiguring the timeout period

The timeout period timer is activated whenever a port encounters a superior BPDU, which then results in a Root Guard violation. If the timeout period is reconfigured while a timer is in use, the timer on that port is set to the new timeout period, minus the time elapsed since the superior BPDU was received.

For example, the original timeout period on a device was configured for 60 seconds. The port encounters a superior BPDU and the timer starts. Issuing a **show span root-protect** command displays the following information.

```
device(config)#show span root-protect
Port    VLAN  Current State
1/4     1     Inconsistent state (56 seconds left on timer)
```

While the timer is in use, the timeout period is changed to 30 seconds through the issue of the following command.

```
device(config)# spanning-tree root-protect timeout 30
```

The timer continues the countdown and minus the time that have already elapsed (about 10 seconds) since the superior BPDU was detected. Issuing a **show span root-protect** command displays the following information.

```
device(config)# show span root-protect
Port    VLAN  Current State
1/4     1     Inconsistent state (20 seconds left on timer)
```

Next, the timeout period is increased to 120 seconds.

```
device(config)# spanning-tree root-protect timeout 120
```

Since the timer has not expired, it continues the countdown. The remaining time left is adjusted by the time that has already elapsed (about 18 seconds) since the superior BPDU was detected. Issuing a **show span root-protect** command displays the following information.

```
device(config)# show span root-protect
Port    VLAN  Current State
1/4     1     Inconsistent state (102 seconds left on timer)
```

Checking for Syslog messages

A Syslog message such as the following is generated after the Root Guard blocks a port.

```
Sep 9 18::39:27:I:STP: Root Guard Port 12/21, VLAN 10 inconsistent (Received superior BPDU)
```

A Syslog message such as the following is generated after the Root Guard unblocks a port.

```
Sep 9 18::39:27:I:STP: Root Guard Port 12/21, VLAN 10 consistent (Timeout)
```

Checking for traps

The following SNMP traps are generated for Root Guard:

- snTrapStpRootGuardDetect is generated after the Root Guard blocks a port.
- snTrapStpRootGuardExpire is generated after a blocked port (due to Root Guard) goes back to a Forwarding state

Refer to the *Unified IP MIB Reference* for details.

BPDU Guard

STP protection provides the ability to prohibit an end station from initiating or participating in an STP topology. The Bridge Protocol Data Units (BPDU) Guard is used to keep all active network topologies predictable.

NOTE
The feature is also available for MSTP and RSTP.

STP detects and eliminates logical loops in a redundant network by selectively blocking some data paths and allowing only some data paths to forward traffic.

In an STP environment, switches, end stations, and other Layer 2 devices use BPDUs to exchange information that STP will use to determine the best path for data flow. When a Layer 2 device is powered ON and connected to the network, or when a Layer 2 device goes down, it sends out an BPDU, triggering a topology change.

In some instances, it is unnecessary for a connected device, such as an end station, to initiate or participate in a topology change. In this case, you can enable the BPDU Guard feature on the Brocade port to which the end station is connected. The BPDU Guard feature disables the connected device's ability to initiate or participate in an topology change, by dropping all BPDUs received from the connected device.

As an extended security measure, the administrator can disable a port if a BPDU is received on a port where BPDU Guard is configured. A Syslog message and SNMP trap are triggered when the port is disabled.

You can re-enable the disabled port from the CLI; however, make sure the offending BPDUs have stopped before re-enabling the port. Otherwise, the port will be disabled again the moment a new BPDU is received.

NOTE
BPDU Guard should be configured only on the primary port of a LAG. If a port configured with BPDU guard is made a secondary port, the LAG deployment will be vetoed.

Enabling BPDU Guard

You can enable BPDU Guard on a per-port basis.

To prevent an end station from initiating or participating in topology changes, enter the following command at the interface level of the CLI.

```
device(config) interface ethe 2/1
device(config-if-e1000-2/1)# spanning-tree protect
```

Syntax: [no] spanning-tree protect

This command causes the port to drop BPDUs sent from the device on the other end of the link.

Enter the **no** form of the command to disable BPDU Guard on the port and remove the **spanning-tree protect do-disable** feature if they are configured.

Enabling BPDU Guard and disabling a port that receives BPDUs

You can enable BPDU Guard on a port and at the same time configure a port to be disabled when it receives a BPDU. Enter the following commands.

```
device(config) interface ethe 2/1
device(config-if-e1000-2/1)#spanning-tree protect do-disable
```

Syntax: [no] spanning-tree protect do-disable

If both **spanning-tree protect** and **spanning-tree protect do-disable** are configured on an interface, **spanning-tree protect do-disable** takes precedence. This means that when the port receives a BPDU, the port will drop the BPDU and disable the port.

If you issue a **no spanning-tree protect do-disable** command, the port will be re-enabled and will no longer be disabled when it receives a BPDU. The following message is displayed when you enter the **no spanning-tree protect do-disable** command.

```
This command removes only "spanning-tree protect do-disable". To remove "spanning-tree protect", please
issue a separate command "no spanning-tree protect".
```

Re-Enabling a port disabled due to BPDU guard

A port disabled by the **spanning-tree protect do-disable** command can be enabled by the following commands:

- Entering the **no spanning-tree protect do-disable** command.
- Entering the **spanning-tree protect re-enable** command. Make sure the offending BPDUs have stopped before issuing this command; otherwise, the port will be disabled again once it receives a new BPDU.

```
device(config)# interface ethernet 1/4
device(config-if-e10000-1/4)#spanning-tree protect re-enable
```

Syntax: [no] spanning-tree protect re-enable

Issuing the **spanning-tree protect re-enable** command does not remove the **spanning-tree protect do-disable** configuration on the port. If a new BPDU is received on the port, the port will be disabled again. To prevent this from happening, you can do one of the following:

- Remove the **spanning-tree protect do-disable** configuration by issuing the **no spanning-tree protect do-disable** command, followed by the **spanning-tree protect re-enable** command to re-enable the port.
- Remove the source of the offending BPDUs from the network.

This command does not have a **no** form.

Displaying BPDU Guard configuration

To determine if BPDU Guard is configured on the device, enter the following command.

```
device#show spanning-tree protect
  protect   Show STP BPDU Guard information
devicer# show span protect
Port    Disable Port on BPDU Rx       Current Port State
1/1     No                            down
1/2     Yes                           down
1/3     No                            up
1/4     Yes                           up
```

Syntax: show spanning-tree protect

The command shows the following information.

TABLE 53 CLI display of show spanning-tree bp

This field...	Displays...
Port	The port on which BPDU Guard is configured
Disable Port on BPDU Rx	Indicates if **spanning-tree protect do-disable** is configured on the port: • Yes - **spanning-tree protect do-disable** is configured on the port. The BPDU will be dropped and the port will be disabled when it receives a BPDU. • No - **spanning-tree protect do-disable** is not configured. The BPDU will be dropped but the port will not be disabled.
Current Port State	Indicates if the port is currently UP or DOWN.

Determining if BPDU Guard is enabled

The **show interface** command displays the state of a port.

If BPDU Guard is disabled or has not been configured, the output shows the following information.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is disabled
```

If BPDU Guard has been enabled using the **spanning-tree protect** command, the output shows the following.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is enabled
```

If BPDU Guard is enabled using the **spanning-tree protect do-disable** command, the output shows.

```
device#show interface ethernet 1/4
10GigabitEthernet1/4 is up, line protocol is up
  STP Root Guard is disabled, STP BPDU Guard is enabled with port to be disabled on BPDU receive
```

Syntax: show interface ethernet slot/port

Checking for Syslog messages

When the **spanning-tree protect do-disable** command is issued, the port becomes disabled and the following Syslog messages are generated.

```
Sep 9 18::39:27:I:STP: BPDU Guard port 1/4 disable
Sep 9 18::39:27:I:System: Interface ethernet 1/4, state down - disabled
```

When the **spanning-tree protect re-enable** command is issued to re-enable a port, the following Syslog messages are generated.

```
Sep 9 18:43:21:I:STP: BPDU Guard re-enabled on ports ethe 1/4
Sep 9 18:43:23:I:System: Interface ethernet 1/4, state up
```

Checking for traps

The following SNMP traps are generated for BPDU Guard. Refer to the *Unified IP MIB Reference* for details:

- snTrapStpBPDUGuardDetect is generated when a port is disabled because **spanning-tree protect do-disable command** on a port and that port received a BPDU and disabled the port.
- snTrapSTPBPDUGuardExpire is generated when a port that has been disabled due to a BPDU Guard violation is re-enabled using the **spanning-tree protect re-enable** command.

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.

```
device# 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
```

To display only ports blocked by the STP protocol, enter the following command at any level of the CLI.

```
Brocade#show spanning-tree blocked vlan 10
VLAN 10 - 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
80000024389e2d00 20     2      15     1     718          1
RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
80000024388f6b00 2         80000024388f6b00 3/1   20  2   15
STP Port Parameters:
Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
3/2   128  2         BLOCKING   0         80000024388f6b00 80000024388f6b00
3/3   128  2         BLOCKING   0         80000024388f6b00 80000024388f6b00
3/4   128  2         BLOCKING   0         80000024388f6b00 80000024388f6b00
```

Syntax: show spanning-tree [**blocked**] [**vlan vlan-id**] | [**pvst-mode**] | **detail** [**vlan vlan-id** [**ethernet slot/port**]] [**begin expression** | **exclude expression** | **include expression**]

The **blocked** parameter displays only ports blocked by the STP protocol. When the **blocked** parameter is not specified, information is displayed for all STP controlled ports.

The **vlan** *vlan-id* parameter displays STP information for the specified port-based VLAN.

The **pvst-mode** parameter displays STP information for the Brocade device's Per VLAN Spanning Tree (PVST+) compatibility configuration. Refer to PVST or PVST+ compatibility on page 356.

The **detail** parameter and its additional optional parameters display detailed information for individual ports. Refer to Displaying detailed STP information for each interface on page 340.

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

TABLE 54 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 54 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 Change sec	The number of seconds since the last time a topology change occurred.
Topology Change cnt	The number of times the topology has changed since this device was reloaded.
Root Bridge Parameters	
Root Identifier	The ID assigned by STP to the root bridge for this spanning tree in hexadecimal.
Root Cost	The cumulative cost from this bridge to the root bridge. If this device is the root bridge, then the root cost is 0.
DesignatedBridge Identifier	The designated bridge to which the root port is connected. The designated bridge is the device that connects the network segment on the port to the root bridge.
Root Port	The port on this device that connects to the root bridge. If this device is the root bridge, then the value is "Root" instead of a port number.
Max Age sec	The number of seconds this root bridge waits for a hello message from the bridges before deciding a bridges has become unavailable and performing a reconvergence.
Hello sec	The interval between each configuration BPDU sent by the root bridge.
FwdDly sec	The number of seconds this root bridge waits following a topology change and consequent reconvergence.
Port STP Parameters	
Port Num	The port number.
Priority	The port's STP priority. **NOTE** If you configure this value, specify it in decimal format. Refer to Changing STP port parameters on page 332.
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.

TABLE 54 CLI display of STP information (continued)

This field...	Displays...
	• DISABLED - The port is not participating in STP. This can occur when the port is disconnected or STP is disabled on the port. • FORWARDING - STP is allowing the port to send and receive frames. • LISTENING - STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING - The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table.
Design Cost	The cost to the root bridge as advertised by the designated bridge that is connected to this port. If the designated bridge is the root bridge itself, then the cost is 0. The identity of the designated bridge is shown in the Design Bridge field.
Designated Root	The root bridge as recognized on this port. The value is the same as the root bridge ID listed in the Root ID field.
Designated Bridge	The bridge as recognized on this port.

Displaying detailed STP information for each interface

To display the detailed STP information, enter the following command at any level of the CLI.

```
device# show spanning-tree detail vlan 10
VLAN 10 - STP instance 1
------------------------------------------------------------------------
STP Bridge Parameters:
Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethernet 1/3 ethernet 1/13
Active global timers - None
STP Port Parameters:
Port 1/3 - DISABLED
Port 1/13 - DISABLED
VLAN 20 - STP instance 2
------------------------------------------------------------------------
STP Bridge Parameters:
Bridge identifier - 0x8000000480a04000
Root bridge - 0x8000000480a04000
Control ports - ethernet 1/3 ethernet 1/13
Active global timers - None
STP Port Parameters:
Port 1/3 - DISABLED
Port 1/13 - DISABLED
```

If a port is disabled, the only information shown by this command is "DISABLED". If a port is enabled, this display shows the following information.

Syntax: show spanning-tree detail [**vlan vlan-id** [**ethernet slot/port**]]

The **vlan** *vlan-id* parameter specifies a VLAN.

The **ethernet** *slot/portnum* parameter specifies an individual port within the VLAN (if specified).

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** command with the **vlan** *vlan-id* parameters 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** command with the *group-id* variable.

The **show spanning-tree detail** command shows the following information for each VLAN participating in the spanning tree.

TABLE 55 CLI display of detailed STP information for ports

This field...	Displays...
VLAN ID	The VLAN that contains the listed ports and the number of STP instances on this VLAN. The STP type can be one of the following: • Proprietary multiple Spanning Tree • IEEE 802.1Q Single Spanning Tree (SSTP) NOTE If STP is disabled on a VLAN, the command displays the following message instead: "Spanning-tree of port-vlan *vlan-id* is disabled."
STP Bridge Parameters:	
Bridge identifier	The STP identity of this device.
Root	The ID assigned by STP to the root bridge for this spanning tree.
Control ports	The ports in the VLAN.
Active global timers	The global STP timers that are currently active, and their current values. The following timers can be listed: • Hello - The interval between Hello packets. This timer applies only to the root bridge. • Topology Change (TC) - The amount of time during which the topology change flag in Hello packets will be marked, indicating a topology change. This timer applies only to the root bridge. • Topology Change Notification (TCN) - The interval between Topology Change Notification packets sent by a non-root bridge toward the root bridge. This timer applies only to non-root bridges.
STP Port Parameters:	
Port number and STP state	The internal port number and the port's STP state. The internal port number is one of the following: • The port's interface number, if the port is the designated port for the LAN. • The interface number of the designated port from the received BPDU, if the interface is not the designated port for the LAN. The state can be one of the following: • BLOCKING - STP has blocked Layer 2 traffic on this port to prevent a loop. The device or VLAN can reach the root bridge using another port, whose state is FORWARDING. When a port is in this state, the port does not transmit or receive user frames, but the port does continue to receive STP BPDUs. • DISABLED - The port is not participating in STP. This can occur when the port is disconnected or STP is administratively disabled on the port.

TABLE 55 CLI display of detailed STP information for ports (continued)

This field...	Displays...
	• FORWARDING - STP is allowing the port to send and receive frames. • LISTENING - STP is responding to a topology change and this port is listening for a BPDU from neighboring bridges in order to determine the new topology. No user frames are transmitted or received during this state. • LEARNING - The port has passed through the LISTENING state and will change to the BLOCKING or FORWARDING state, depending on the results of STP's reconvergence. The port does not transmit or receive user frames during this state. However, the device can learn the MAC addresses of frames that the port receives during this state and make corresponding entries in the MAC table. NOTE If the state is DISABLED, no further STP information is displayed for the port.

IEEE Single Spanning Tree (SSTP)

By default, each port-based VLAN on the Brocade device runs a separate spanning tree, which you can enable or disable on an individual VLAN basis.

Alternatively, you can configure the Brocade device to run a single spanning tree across all of its ports and VLANs. The SSTP feature is especially useful for connecting a Brocade device to third-party devices that run a single spanning tree in accordance with the 802.1q specification.

SSTP uses the same parameters, with the same value ranges and defaults, as the default STP supported on the Brocade device. Refer to Default STP bridge and port parameters on page 331.

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 Brocade device places all the ports in a non-configurable VLAN, 4095, to implement the SSTP domain. However, this VLAN does not affect port membership in the port-based VLANs you have configured. Other broadcast traffic is still contained within the individual port-based VLANs. Therefore, you can use SSTP while still using your existing VLAN configurations without changing your network. In addition, SSTP does not affect 802.1q tagging. Tagged and untagged ports alike can be members of the single spanning tree domain.

NOTE
When SSTP is enabled, the BPDUs on tagged ports go out untagged.

If you disable SSTP, all VLANs that were members of the single spanning tree now do not run any form of spanning tree. Per VLAN STP can be enabled again using either global STP enable or the spanning-tree enable command under individual VLANs.

NOTE
If the Brocade device has only one port-based VLAN (the default VLAN), then it is already running a single instance of STP. In this case, you do not need to enable SSTP. You need to enable SSTP only if the Brocade device contains more than one port-based VLAN and you want all the ports to be in the same STP broadcast domain.

To configure the Brocade device to run a single spanning tree, enter the following command at the global CONFIG level.

```
device(config)# spanning-tree single
```

To change a global STP parameter, enter a command such as the following at the global CONFIG level.

```
device(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.

```
device(config) spanning-tree single ethernet 1/1 priority 10
```

The commands shown above override the global setting for STP priority and set the priority to 10 for port 1/1.

Here is the syntax for the global STP parameters:

Syntax: [no] spanning-tree single [**forward-delay value**] [**hello-time value**] | [**maximum-age time**] | [**priority value**]

Here is the syntax for the STP port parameters:

Syntax: [no] spanning-tree single [**ethernet slot/portnum path-cost value** | **priority value**]

For the parameter definitions and possible values, refer to Default STP bridge and port parameters on page 331.

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.

Displaying SSTP information

To verify that SSTP is in effect, enter the following commands at any level of the CLI.

```
device(config)# show spanning-tree
VLAN 4095 - STP instance 0
------------------------------------------------------------------------
STP Bridge Parameters:
Bridge           Bridge Bridge Bridge Hold  LastTopology Topology
Identifier       MaxAge Hello  FwdDly Time  Change       Change
hex              sec    sec    sec    sec   sec          cnt
8000000480a04000 20     2      15     1     0            0
RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
8000000480a04000 0         8000000480a04000 Root  20  2   15
STP Port Parameters:
Port  Prio Path      State      Designat- Designated       Designated
Num   rity Cost                ed Cost   Root             Bridge
1/3   128  4         DISABLED   0         0000000000000000 0000000000000000
1/13  128  4         DISABLED   0         0000000000000000 0000000000000000
SSTP members: 10 20 30 99 to 100
```

For information on the command syntax, refer to Displaying STP information on page 337.

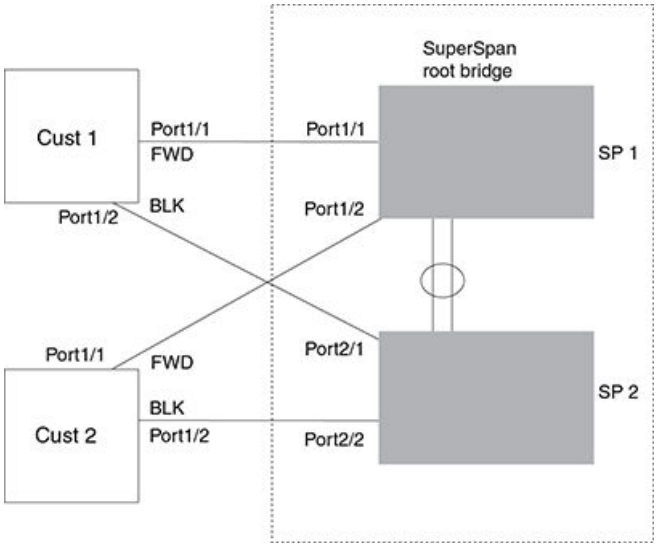
SuperSpan™

SuperSpan is a Brocade STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks. The SP devices are Brocade devices and are configured to tunnel each customer's STP BPDUs through the SP. From the customer's perspective, the SP network is a loop-free non-blocking device or network. The SP network behaves like a hub in the sense that the necessary blocking occurs in the customer network, not in the SP.

The interfaces that connect the SP to a customer's network are configured as SuperSpan boundary interfaces. Each SuperSpan boundary interface is configured with a customer ID, to uniquely identify the customer's network within SuperSpan.

Figure 65 shows an example SuperSpan implementation. In this example, an SP's network is connected to multiple customers. Each customer network is running its own instance of standard STP. The Brocade devices in the SP are running SuperSpan.

FIGURE 63 SuperSpan example



In this example, the SP network contains two devices that are running SuperSpan. The SP is connected to two customer networks. Each customer network is running its own instance of STP. SuperSpan prevents Layer 2 loops in the traffic flow with each customer while at the same time isolating each customer's traffic and spanning tree from the traffic and spanning trees of other customers. For example, the SP devices provide loop prevention for Customer 1 while ensuring that Customer 1's traffic is never forwarded to Customer 2. In this example, customer 1 has two interfaces to the SP network, ports 1/1 and 1/2 connected to SP 1. The SP network behaves like a non-blocking hub. BPDUs are tunneled through the network. To prevent a Layer 2 loop, customer 1's port 1/2 enters the blocking state.

Customer ID

SuperSpan uses a SuperSpan customer ID to uniquely identify and forward traffic for each customer. You assign the customer ID as part of the SuperSpan configuration of the Brocade devices in the SP. In SuperSpan™ on page 344, 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 Brocade device receives a customer's BPDU on a boundary interface, the Brocade device changes the destination MAC address of the BPDU from the bridge group address (00-00-00-00-00-00) 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: 00-00-00-00-01-00.

Each Brocade device that is configured for SuperSpan forwards the BPDU using the changed destination MAC address. At the other end of the tunnel, the Brocade device connected to the customer's network changes the destination MAC address back to the bridge group address (00-00-00-00-00-00).

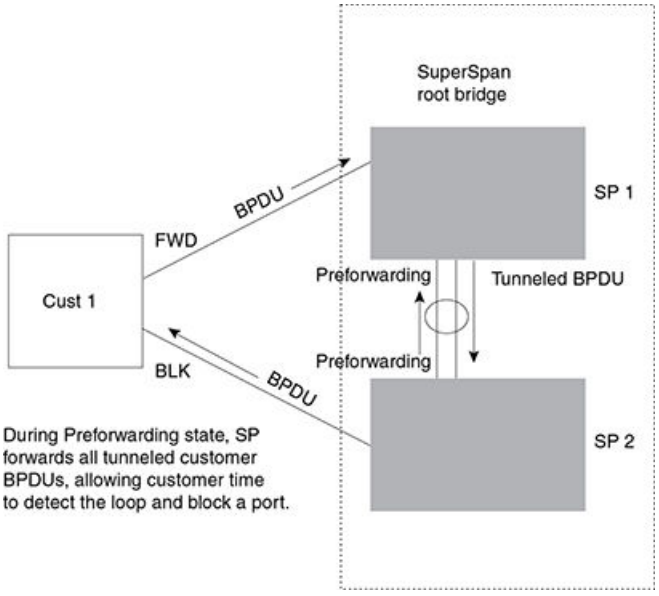
Preforwarding state

To ensure that the customer's network has time to converge at Layer 2 and prevent loops, the Brocade devices configured for SuperSpan use a special forwarding state, Preforwarding. The Preforwarding state occurs between the Learning and Forwarding states and by default lasts for five seconds. During the Preforwarding state, the Brocade device forwards tunneled BPDUs from customers only and does not forward data traffic. This ensures that the customer's network will detect the Layer 2 loop and block a port. The SP network remains unblocked. After the Preforwarding state, the 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 66 shows an example of how the Preforwarding state is used.

FIGURE 64 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 Brocade device to a network that is already running SuperSpan, you must enable SuperSpan on the Brocade device, at least on the VLANs that will be tunneling the customer traffic. Otherwise, the new Brocade device 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.

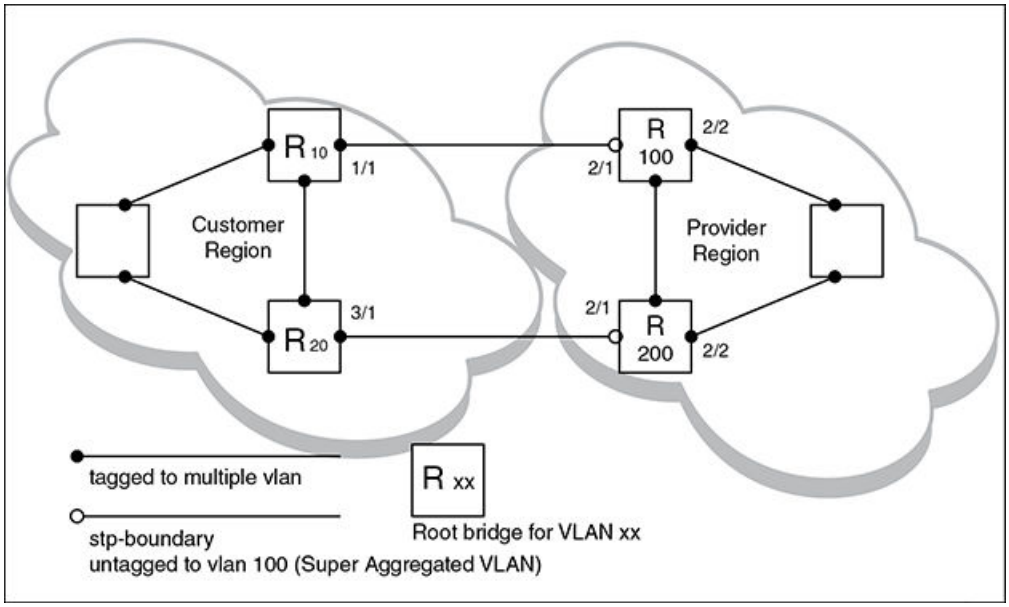
The following sections provide an example of each combination.

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 67 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 65 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.

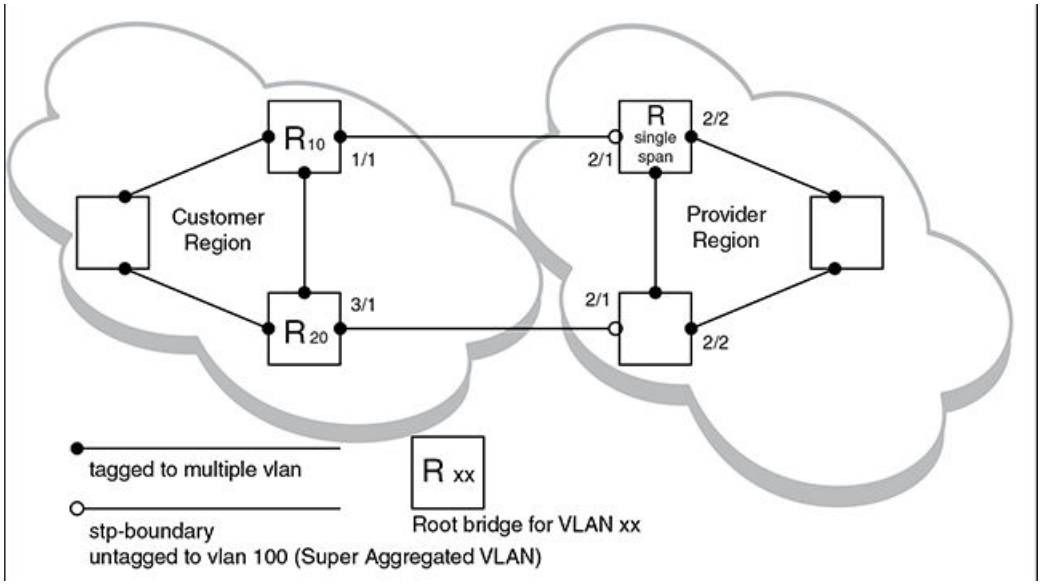
In the above example, STP in VLAN 10 will select R10 as the root bridge and make 1/1 on R10 forwarding while blocking port 3/1 on R20. The opposite occurs for STP in VLAN 20. As a result, both links connecting the customer and SP regions are fully utilized and serve as backup links at the same time, 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 68 shows an example of SuperSpan where the customer network uses multiple spanning trees while the SP network uses Single STP.

FIGURE 66 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.

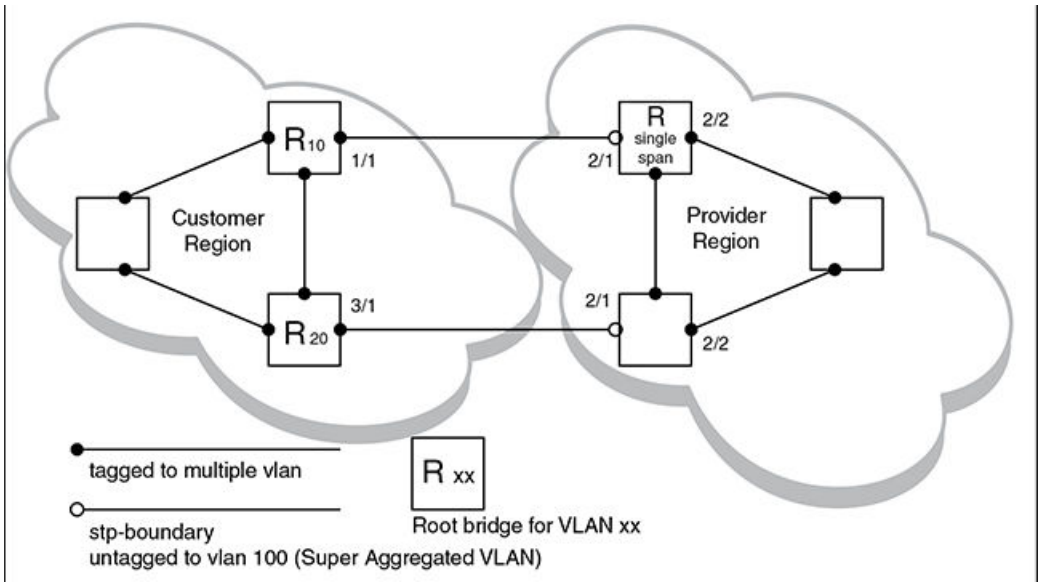
Traffic from all VLANs in the customer network will be aggregated through VLAN 100 at the SP. This setup leaves the customer network's switching pattern virtually unchanged from the scenario in Customer and SP use multiple spanning trees on page 346, since

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 69 shows an example of SuperSpan where the customer network uses Single STP while the SP uses multiple spanning trees.

FIGURE 67 Customer using Single STP and SP using multiple spanning trees

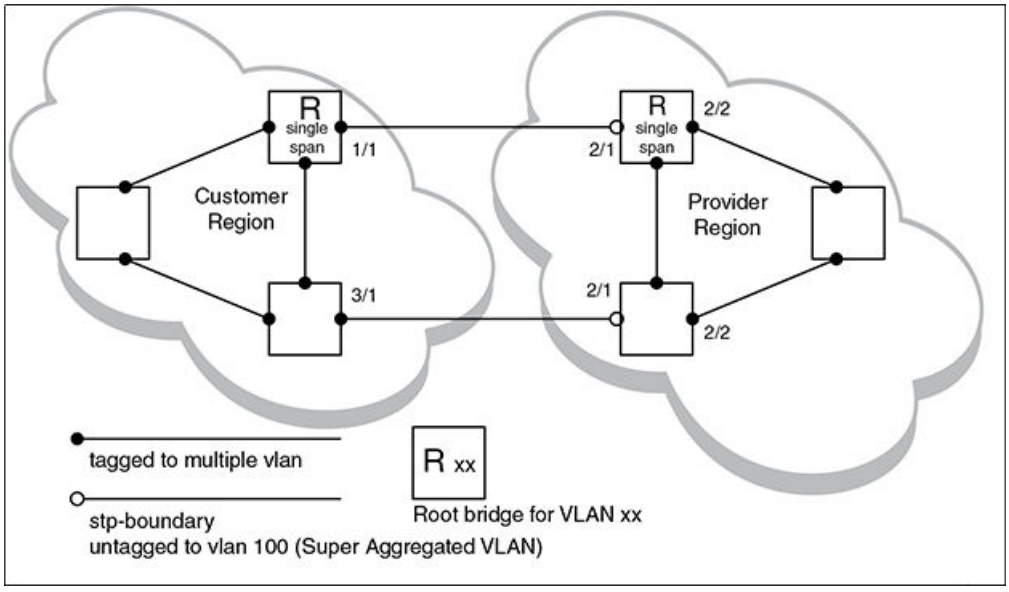


In this setup, the customer network is running a single spanning tree for VLANs 10 and 20. The traffic from VLAN 10 and 20 will be carried, or aggregated by VLAN 100 at the SP's network. The main difference between this scenario and the previous tow scenarios is that all traffic at the customer's network 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 70 shows an example of SuperSpan where the customer network and SP both use Single STP.

FIGURE 68 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 Brocade device for SuperSpan:

- Configure each interface on the Brocade device that is connected to customer equipment as a boundary interface. This step enables the interface to convert the destination MAC address in the customer's BPDUs.

The software requires you to specify a SuperSpan customer ID when configuring the boundary interface. Use an ID from 1 - 65535. The customer ID uniquely identifies the customer. Use the same customer ID for each SP interface with the same customer. When tunneling BPDUs through the network, the Brocade 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 SuperSpan™ on page 344, enter the following commands.

```
device(config)# interface 1/1
device(config-if-e1000-1/1)# stp-boundary 1
device(config)# interface 1/2
device(config-if-e1000-1/2)# stp-boundary 2
```

These commands configure two interfaces on the Brocade device as SuperSpan boundary interfaces. Interface 1/1 is a boundary interface with customer 1. Interface 1/2 is a boundary interface with customer 2. Each boundary interface is associated with a number, which is the SuperSpan ID. The SuperSpan ID identifies the instance of SuperSpan you are associating with the interface. Use the same SuperSpan ID for each boundary interface with the same customer. Use a different SuperSpan ID for each customer. For example, use SuperSpan ID 1 for all the boundary interfaces with customer 1 and use SuperSpan ID 2 for all boundary interfaces with customer 2.

Syntax: [no] stp-boundary num

The *num* parameter specifies the SuperSpan ID. Possible values: 1 - 65535.

To configure the boundary interfaces on SP 2 in SuperSpan™ on page 344, enter the following commands.

```
device(config)# interface 2/1
device(config-if-e1000-2/1)# stp-boundary 1
device(config)# interface 2/2
device(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.

```
device(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 Brocade device. To disable SuperSpan in an individual VLAN, enter commands such as the following.

```
device(config)# vlan 10
device(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.

```
device(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 Brocade device has two SuperSpan customer IDs.

Syntax: show superspan [**cid num**]

The **cid***num* parameter specifies a SuperSpan customer ID. If you do not specify a customer ID, information for all the customer IDs configured on the Brocade device is shown.

This command shows the following information.

TABLE 56 CLI display of SuperSpan customer ID information

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

To display general STP information, refer to Displaying STP information on page 337.

STP feature configuration

Spanning Tree Protocol (STP) features extend the operation of standard STP, enabling you to fine- tune standard STP and avoid some of its limitations.

This section describes how to configure these parameters using the CLI.

Fast port span

When STP is running on a device, message forwarding is delayed during the spanning tree recalculation period following a topology change. The STP forward delay parameter specifies the period of time a bridge waits before forwarding data packets. The forward delay controls the listening and learning periods of STP reconvergence. You can configure the forward delay to a value from 4 - 30 seconds. The default is 15 seconds. Thus, using the standard forward delay, convergence requires 30 seconds (15 seconds for listening and an additional 15 seconds for learning) when the default value is used.

This slow convergence is undesirable and unnecessary in some circumstances. The Fast Port Span feature allows certain ports to enter the forwarding state in four seconds. Specifically, Fast Port Span allows faster convergence on ports that are attached to end stations and thus do not present the potential to cause Layer 2 forwarding loops. Because the end stations cannot cause forwarding loops, they can safely go through the STP state changes (blocking to listening to learning to forwarding) more quickly than is allowed by the standard STP convergence time. Fast Port Span performs the convergence on these ports in four seconds (two seconds for listening and two seconds for learning).

In addition, Fast Port Span enhances overall network performance in the following ways:

- Fast Port Span reduces the number of STP topology change notifications on the network. When an end station attached to a Fast Span port comes up or down, the Brocade device does not generate a topology change notification for the port. In this situation, the notification is unnecessary since a change in the state of the host does not affect the network topology.
- Fast Port Span eliminates unnecessary MAC cache aging that can be caused by topology change notifications. Bridging devices age out the learned MAC addresses in their MAC caches if the addresses are not refreshed for a given period of time, sometimes called the MAC aging interval. When STP sends a topology change notification, devices that receive the notification use the value of the STP forward delay to quickly age out their MAC caches. For example, if a device normal MAC aging interval is 5 minutes, the aging interval changes temporarily to the value of the forward delay (for example, 15 seconds) in response to an STP topology change.

NOTE
While running STP, the devices set the MAC age to forward delay value when these devices receive BPDUs with TC flag set. Since the default value of forward delay is 15 seconds, it is set to 15 seconds for MAC aging. For 20x10G, 2x100G(half slot), 4x10-4x1G (IPSec security) interface modules, the MAC aging value cannot be set to less than 20 seconds. Hence for any forward delay value less than or equal to 15 seconds, the MAC aging value is set to 20 seconds.

In normal STP, the accelerated cache aging occurs even when a single host goes up or down. Because Fast Port Span does not send a topology change notification when a host on a Fast Port Span port goes up or down, the unnecessary cache aging that can occur in these circumstances under normal STP is eliminated.

Fast Port Span is a system-wide parameter and is enabled by default. Thus, when you boot a device, all the ports that are attached only to end stations run Fast Port Span. For ports that are not eligible for Fast Port Span, such as ports connected to other networking devices, the device automatically uses the normal STP settings. If a port matches any of the following criteria, the port is ineligible for Fast Port Span and uses normal STP instead:

- The port is 802.1Q tagged
- The port is a member of a trunk group
- The port has learned more than one active MAC address
- An STP Configuration BPDU has been received on the port, thus indicating the presence of another bridge on the port.

You also can explicitly exclude individual ports from Fast Port Span if needed. For example, if the only uplink ports for a wiring closet switch are Gbps ports, you can exclude the ports from Fast Port Span.

Disabling and re-enabling fast port span

Fast Port Span is a system-wide parameter and is enabled by default. Therefore, all ports that are eligible for Fast Port Span use it.

To disable or re-enable Fast Port Span, enter the following commands.

```
device(config)#no fast port-span
device(config)#write memory
```

Syntax: [no] fast port-span

NOTE
The **fast port-span** command has additional parameters that let you exclude specific ports. These parameters are shown in the following section.

To re-enable Fast Port Span, enter the following commands.

```
device(config)#fast port-span
device(config)#write memory
```

Excluding specific ports from fast port span

To exclude a port from Fast Port Span while leaving Fast Port Span enabled globally, enter commands such as the following.

```
device(config)#fast port-span exclude ethernet 1
device(config)#write memory
```

To exclude a set of ports from Fast Port Span, enter commands such as the following.

```
device(config)#fast port-span exclude ethernet 1 ethernet 2 ethernet 3
device(config)#write memory
```

To exclude a contiguous (unbroken) range of ports from Fast Span, enter commands such as the following.

```
device(config)#fast port-span exclude ethernet 1 to 24
device(config)#write memory
```

Syntax: [no] fast port-span [**exclude ethernet** *port* [**ethernet** *port*] | **to** [*port*]]

Specify the port variable in one of the following formats:

- FWS, FCX, and ICX stackable switches - stack-unit/slotnum/portnum
- FSX 800 and FSX 1600 chassis devices - slotnum/portnum
- ICX devices - slotnum/portnum
- FESX compact switches - portnum

To re-enable Fast Port Span on a port, enter a command such as the following.

```
device(config)#no fast port-span exclude ethernet 1
device(config)#write memory
```

This command re-enables Fast Port Span on port 1 only and does not re-enable Fast Port Span on other excluded ports. You also can re-enable Fast Port Span on a list or range of ports using the syntax shown above this example.

To re-enable Fast Port Span on all excluded ports, disable and then re-enable Fast Port Span by entering the following commands.

```
device(config)#no fast port-span
device(config)#fast port-span
device(config)#write memory
```

Disabling and then re-enabling Fast Port Span clears the exclude settings and thus enables Fast Port Span on all eligible ports. To make sure Fast Port Span remains enabled on the ports following a system reset, save the configuration changes to the startup-config file after you re-enable Fast Port Span. Otherwise, when the system resets, those ports will again be excluded from Fast Port Span.

Fast Uplink Span

The Fast Port Span feature described in the previous section enhances STP performance for end stations. The Fast Uplink Span feature enhances STP performance for wiring closet switches with redundant uplinks. Using the default value for the standard STP forward delay, convergence following a transition from an active link to a redundant link can take 30 seconds (15 seconds for listening and an additional 15 seconds for learning).

You can use the Fast Uplink Span feature on a Brocade device deployed as a wiring closet switch to decrease the convergence time for the uplink ports to another device to just one second. The new Uplink port directly goes to forward mode (bypassing listening and learning modes). The wiring closet switch must be a Brocade device but the device at the other end of the link can be a Brocade device or another vendor's switch.

Configuration of the Fast Uplink Span feature takes place entirely on the Brocade device. To configure the Fast Uplink Span feature, specify a group of ports that have redundant uplinks on the wiring closet switch (Brocade device). If the active link becomes unavailable, the Fast Uplink Span feature transitions the forwarding to one of the other redundant uplink ports in just one second. All Fast Uplink Span-enabled ports are members of a single Fast Uplink Span group.

> NOTE
> To avoid the potential for temporary bridging loops, Brocade recommends that you use the Fast Uplink feature only for wiring closet switches (switches at the edge of the network cloud). In addition, enable the feature only on a group of ports intended for redundancy, so that at any given time only one of the ports is expected to be in the forwarding state.

NOTE
When the wiring closet switch (Brocade device) first comes up or when STP is first enabled, the uplink ports still must go through the standard STP state transition without any acceleration. This behavior guards against temporary routing loops as the switch tries to determine the states for all the ports. Fast Uplink Span acceleration applies only when a working uplink becomes unavailable.

Active uplink port failure

The active uplink port is the port elected as the root port using the standard STP rules. All other ports in the group are redundant uplink ports. If an active uplink port becomes unavailable, Fast Uplink Span transitions the forwarding of traffic to one of the redundant ports in the Fast Uplink Span group in one second bypassing listening and learning port states.

Switchover to the active uplink port

When a failed active uplink port becomes available again, switchover from the redundant port to the active uplink port is delayed by 30 seconds. The delay allows the remote port to transition to forwarding mode using the standard STP rules. After 30 seconds, the blocked active uplink port begins forwarding in just one second and the redundant port is blocked.

NOTE
Use caution when changing the spanning tree priority. If the switch becomes the root bridge, Fast Uplink Span will be disabled automatically.

Fast Uplink Span Rules for Trunk Groups

If you add a port to a Fast Uplink Span group that is a member of a trunk group, the following rules apply:

- If you add the primary port of a trunk group to the Fast Uplink Span group, all other ports in the trunk group are automatically included in the group. Similarly, if you remove the primary port in a trunk group from the Fast Uplink Span group, the other ports in the trunk group are automatically removed from the Fast Uplink Span group.
- You cannot add a subset of the ports in a trunk group to the Fast Uplink Span group. All ports in a trunk group have the same Fast Uplink Span property, as they do for other port properties.
- If the working trunk group is partially down but not completely down, no switch-over to the backup occurs. This behavior is the same as in the standard STP feature.
- If the working trunk group is completely down, a backup trunk group can go through an accelerated transition only if the following are true:
  - The trunk group is included in the fast uplink group.
  - All other ports except those in this trunk group are either disabled or blocked. The accelerated transition applies to all ports in this trunk group.

When the original working trunk group comes back (partially or fully), the transition back to the original topology is accelerated if the conditions listed above are met.

Configuring a Fast Uplink Port Group

To configure a group of ports for Fast Uplink Span, enter the following commands:

```
Brocade(config)# fast uplink-span ethernet 4/1 to 4/4
Brocade(config)# write memory
```

Syntax: [no] fast uplink-span [**ethernet** *port* [**ethernet** *port* ... | **to** *port*]]

Specify the port variable in one of the following formats:

- FWS, FCX, and ICX stackable switches - stack-unit/slotnum/portnum
- FSX 800 and FSX 1600 chassis devices - slotnum/portnum
- ICX devices - slotnum/portnum
- FESX compact switches - portnum

This example configures four ports, 4/1 - 4/4, as a Fast Uplink Span group. In this example, all four ports are connected to a wiring closet switch. Only one of the links is expected to be active at any time. The other links are redundant. For example, if the link on port 4/1 is the active link on the wiring closet switch but becomes unavailable, one of the other links takes over. Because the ports are configured in a Fast Uplink Span group, the STP convergence takes one second instead of taking atleast 30 seconds using the standard STP forward delay.

You can add ports to a Fast Uplink Span group by entering the fast uplink-span command additional times with additional ports. The device can have only one Fast Uplink Span group, so all the ports you identify as Fast Uplink Span ports are members of the same group.

To remove a Fast Uplink Span group or to remove individual ports from a group, use "no" in front of the appropriate fast uplink-span command. For example, to remove ports 4/3 and 4/4 from the Fast Uplink Span group configured above, enter the following commands:

```
Brocade(config)# no fast uplink-span ethernet 4/3 to 4/4
Brocade(config)# write memory
```

To check the status of ports with Fast Uplink Span enabled.

```
Brocade(config)# show span fast-uplink-span
STP instance owned by VLAN 1
Global STP (IEEE 802.1D) Parameters:
VLAN Root             Root Root   Prio Max He- Ho- Fwd Last    Chg Bridge
 ID   ID              Cost Port   rity Age llo ld  dly Chang   cnt Address
                                  Hex  sec sec sec sec sec
   1 000000c100000001 2    1/3/1  8000 20  2   1   15  65      15  000011111111
Port STP Parameters:
Port   Prio Path  State       Fwd    Design  Designated       Designated
Num    rity Cost              Trans  Cost    Root             Bridge
       Hex
1/1/2  80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/1/3  80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/1/4  80   4     FORWARDING  1      2       000000c100000001 8000000011111111
1/1/5  80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/1/6  80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/1/7  80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/1/8  80   0     DISABLED    0      0       0000000000000000 0000000000000000
1/1/9  80   0     DISABLED    0      0       0000000000000000 0000000000000000
```

Syntax: show span fast-uplink-span

Configuring Fast Uplink Span within a VLAN

You can also configure Fast Uplink Span on the interfaces within a VLAN.

To configure Fast Uplink Span for a VLAN, enter command such as the following.

```
device(config)#vlan 10
device(config-vlan-10)#untag ethernet 8/1 to 8/2
device(config-vlan-10)#fast uplink-span ethernet 8/1 to 8/2
```

Syntax: [no] fast uplink-span ethernet *port-no*

To check the status of Fast Uplink Span for a specified VLAN.

```
Brocade(config-vlan-2)#show span vlan 2 fast-uplink-span
STP instance owned by VLAN 2
```

```
Global STP (IEEE 802.1D) Parameters:
VLAN Root             Root Root   Prio Max He- Ho- Fwd Last    Chg Bridge
 ID   ID              Cost Port   rity Age llo ld  dly Chang   cnt Address
                                  Hex  sec sec sec sec sec
   2 8000000011111111 0    Root   8000 20  2   1   15  29596   0   000011111111
Port STP Parameters:
Port   Prio Path  State       Fwd    Design  Designated       Designated
Num    rity Cost              Trans  Cost    Root             Bridge
       Hex
1/1/1  80   4     LISTENING   0      0       8000000011111111 8000000011111111
```

Syntax: show span vlan *vlan-id* **fast-uplink-span**

The **vlan** *vlan-id* parameter displays Fast Uplink Span information for the specified VLAN.

Configuring STP under an ESI VLAN

STP can also be configured under a VLAN that is part of a user-configured ESI. For example, to enable spanning tree on a VLAN that is part of an ESI, configure the following commands.

```
device(config)# esi customer1 encapsulation cvlan
device(config-esi-customer1)# vlan 100
device(config-esi-customer1-vlan-100)# spanning-tree
```

Configuration considerations:

The configuration considerations are as follows:

- MSTP can only be configured under the default ESI. MSTP cannot be configured for VLANs that are configured under a user-defined ESI.
- STP can be configured for VLANs with encapsulation type B-VLAN, S-VLAN or C-VLAN.

When STP or RSTP is configured for VLANs under an ESI, the MRP members must be part of the same ESI.

PVST or PVST+ compatibility

Brocade's support for Cisco's Per VLAN Spanning Tree plus (PVST+) allows the Brocade device to run multiple spanning trees (MSTP) while also interoperating with IEEE 802.1Q devices. Ports automatically detect PVST+ BPDUs and enable support for the BPDUs once detected.

When it is configured for MSTP, the Brocade device can interoperate with PVST.

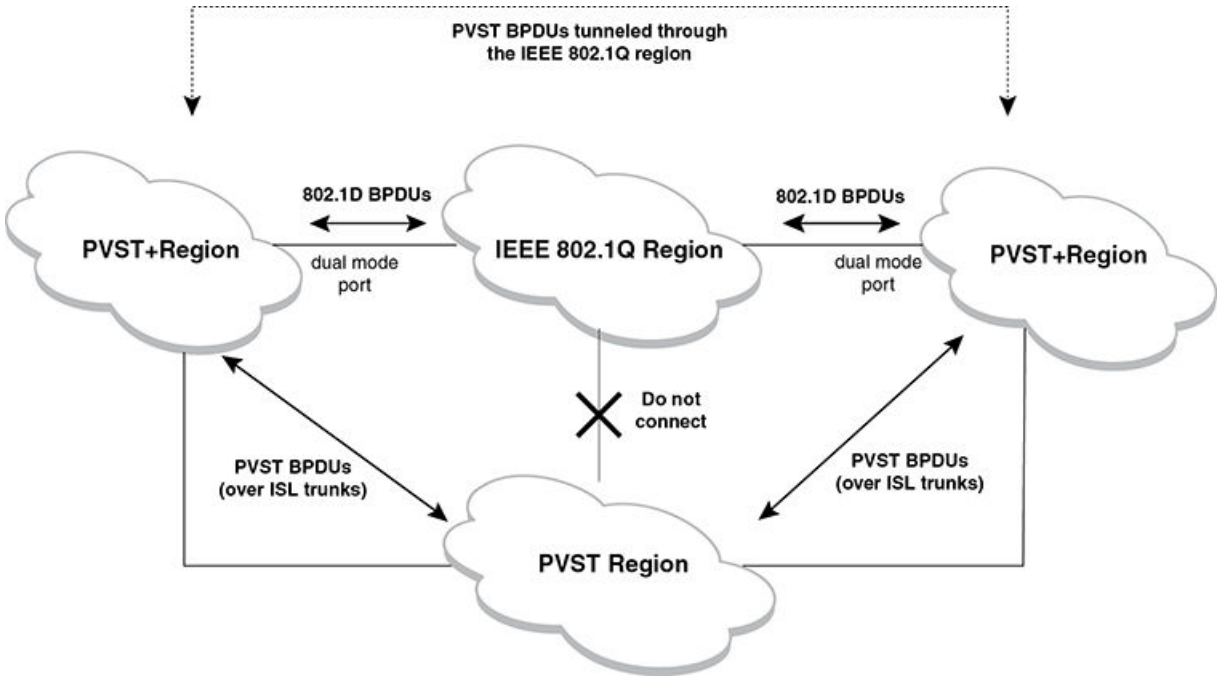
Overview of PVST and PVST+

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 Brocade device to interoperate with PVST spanning trees and the IEEE 802.1Q spanning tree at the same time.

IEEE 802.1Q and PVST regions cannot interoperate directly but can interoperate indirectly through PVST+ regions. PVST BPDUs are tunneled through 802.1Q regions, while PVST BPDUs for VLAN 1 (the IEEE 802.1Q VLAN) are processed by PVST+ regions. Figure 71 shows the interaction of IEEE 802.1Q, PVST, and PVST+ regions.

FIGURE 69 Interaction of IEEE 802.1Q, PVST, and PVST+ regions



VLAN Tags and dual mode

The **dual-mode** feature enables the port to send and receive both tagged and untagged frames on a port. When the dual-mode feature is enabled, the port is an untagged member of one of its VLANs and is at the same time a tagged member of all its other VLANs. The untagged frames are supported on the port's **Port Native VLAN** .

To interoperate with other vendors, the dual-mode feature must be enabled on the port. Some vendors use VLAN 1 by default to support the IEEE 802.1Q based standard spanning tree protocols such as 802.1d and 802.1w for sending the untagged frames on VLAN 1. On Brocade devices by default, the Port Native VLAN is the same as the device's **Default VLAN1** , which by default is VLAN 1. Thus, to support IEEE 802.1Q in a typical configuration, the port must be able to send and receive untagged frames for VLAN 1 and tagged frames for the other VLANs and interoperate with the vendors also using VLAN 1. If you want to use tagged frames on VLAN 1, you can change the default VLAN ID to an ID other than 1. You also can specify the VLAN on which you want the port to send and receive untagged frames (the Port Native VLAN). The Port Native VLAN ID does not need to be the same as the Default VLAN. Make sure that untagged (Native) VLAN is also changed on the interoperating vendor side to match with that on the Brocade side.

To support the IEEE 802.1Q with non-standard proprietary protocols such as PVST and PVST+, a port must always send and receive untagged frames on VLAN 1 on both sides. In that case, enable the dual-mode 1 feature to allow untagged BPDUs on VLAN 1and use Native VLAN 1 on the interoperating vendor side. You should not use VLAN 1 for tagged frames in this case.

NOTE
Support for the IEEE 802.1Q spanning tree always uses VLAN 1, regardless of whether the Brocade 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.

The tagged port also supports IEEE 802.1Q BPDUs, since the dual-mode feature on the port is enabled, by default.

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 interoperable with PVST+ to revert to multiple spanning tree when connected to a Brocade device.

Enabling PVST+ support manually

To immediately enable PVST+ support on a port, enter commands such as the following.

```
device(config)# interface ethernet 1/1
device(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 Brocade device, enter the following command at any level of the CLI.

```
device(config)# show span pvst-mode
PVST+ Enabled on:
Port      Method
1/1       Set by configuration
1/2       Set by configuration
2/10      Set by auto-detect
3/12      Set by configuration
4/24      Set by auto-detect
```

Syntax: show span pvst-mode

This command displays the following information.

TABLE 57 CLI display of PVST+ information

This field...	Displays...
Port	The port number. **NOTE** The command lists information only for the ports on which PVST+ support is enabled.
Method	The method by which PVST+ support was enabled on the port. The method can be one of the following: • Set by configuration - You enabled the support. • Set by auto-detect - The support was enabled automatically when the port received a PVST+ BPDU.

Configuration examples

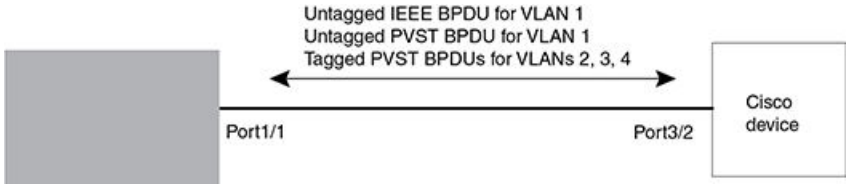
The examples use two common configurations:

- Untagged IEEE 802.1Q BPDUs on VLAN 1 and tagged PVST+ BPDUs on other VLANs
- Tagged IEEE 802.1Q BPDUs on VLAN 1 and untagged BPDUs on another VLAN

Tagged port using default VLAN 1 as its port native VLAN

In Figure 72, a PVST+ configuration uses VLAN 1 as the untagged default VLAN and VLANs 2, 3, and 4 as tagged VLANs.

FIGURE 70 Default VLAN 1 for untagged BPDUs



To implement this configuration, enter the following commands on the Brocade device.

```
device(config)# vlan-group 1 vlan 2 to 4
device(config-vlan-group-1)# tagged ethernet 1/1
device(config-vlan-group-1)# exit
device(config)# interface ethernet 1/1
device(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.

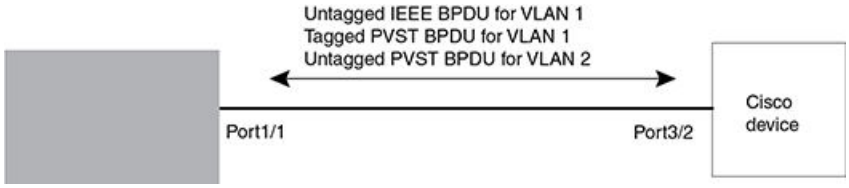
Port 1/1 will process BPDUs as follows:

- 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 73, 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 71 Port Native VLAN 2 for untagged BPDUs



To implement this configuration, enter the following commands on the Brocade device.

```
device(config)# default-vlan-id 4000
device(config)# vlan 1
device(config-vlan-1)# tagged ethernet 1/1
device(config-vlan-1)# exit
device(config)# vlan 2
device(config-vlan-2)# untagged ethernet 1/1
device(config-vlan-2)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
device(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.

```
device(config)# default-vlan-id 1000
device(config)# vlan 1
device(config-vlan-1)# tagged ethernet 1/1 to 1/2
device(config-vlan-1)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
device(config-if-e10000-1/1)# exit
device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# pvst-mode
device(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.

```
device(config)# default-vlan-id 1000
device(config)# vlan 1
device(config-vlan-1)# tagged ethernet 1/1 to 1/2
device(config-vlan-1)# exit
```

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# pvst-mode
device(config-if-e10000-1/1)# exit
device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# pvst-mode
device(config-if-e10000-1/2)# exit
```

Setting the ports as dual-mode ensures that the untagged IEEE 802.1Q BPDUs reach the VLAN 1 instance.

802.1s Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP) as defined in IEEE 802.1s allows you to configure multiple STP instances. This will allow several VLANs to be mapped to a reduced number of spanning-tree instances. This ensures loop-free topology for 1 or more VLANs that have the same Layer 2 topology.

> NOTE
> In addition to the features described in this chapter, Root Guard and BPDU Guard are supported. Refer to Root Guard on page 332 and BPDU Guard on page 334 for details.

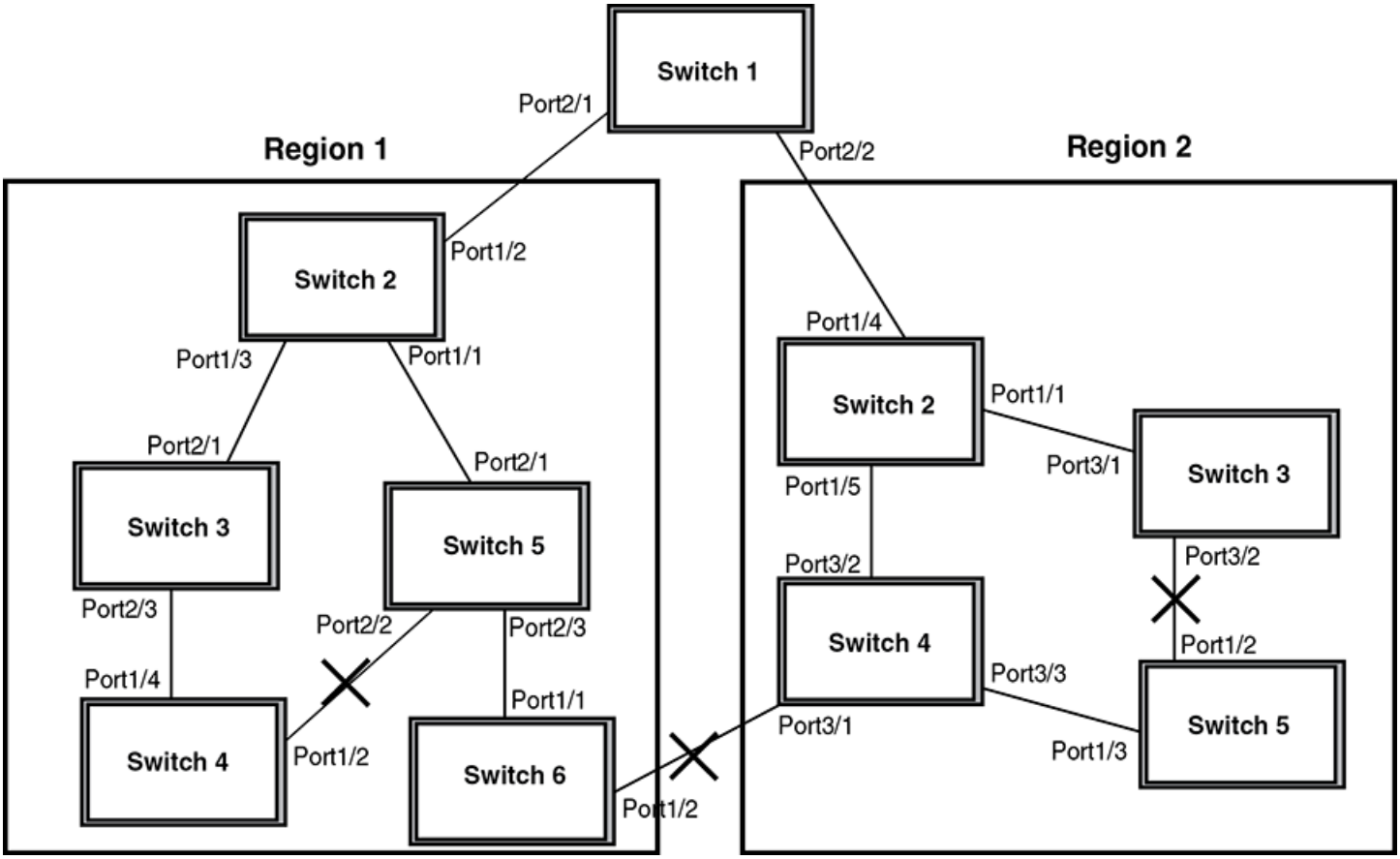
Multiple Spanning-Tree regions

Using MSTP, the entire network runs a common instance of RSTP. Within that common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the common spanning tree instance (CST) and the regions run a local instance. The local instance is known as Internal Spanning Tree (IST). The CST treats each instance of IST as a single bridge. Consequently, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. In addition, MSTP can coexist with individual devices running STP or RSTP in the Common and Internal Spanning Trees instance (CIST). With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

For example, in Figure 74 a network is configured with two regions: Region 1 and Region 2. The entire network is running an instance of CST. Each of the regions is running an instance of IST. In addition, this network contains Switch 1 running RSTP that is not configured in a region and, consequently, is running in the CIST instance. In this configuration, the regions are each regarded as a single bridge to the rest of the network, as is Switch 1. The CST prevents loops from occurring across the network. Consequently, a port is blocked at port 1/2 of Switch 4.

Additionally, loops must be prevented in each of the IST instances. Within the IST Region 1, a port is blocked at port 1/2 of Switch 4 to prevent a loop in that region. Within Region 2, a port is blocked at port 3/2 of Switch 3 to prevent a loop in that region.

FIGURE 72 MSTP configured network



The following definitions describe the STP instances that define an MSTP configuration:

Common Spanning (CST) - MSTP runs a single instance of spanning tree, called the Common Spanning Tree (CST), across all the bridges in a network. This instance treats each region as a single bridge. It all other ways, it operates exactly like Rapid Spanning Tree (RSTP).

Internal Spanning Tree (IST) - Instances of spanning tree that operate within a defined region are called ISTs (Internal Spanning Tree).

Common and Internal Spanning Trees (CIST) - This is the default MSTP instance 0. It contains all of the ISTs and all bridges that are not formally configured into a region. This instance interoperates with bridges running legacy STP and RSTP implementations.

Multiple Spanning Tree Instance (MSTI) - The MSTI is identified by an MST identifier (MSTid) value between 1 and 4094. This defines an individual instance of an IST. One or more VLANs can be assigned to an MSTI. A VLAN cannot be assigned to multiple MSTIs.

MSTP Region - These are clusters of bridges that run multiple instances of the MSTP protocol. Multiple bridges detect that they are in the same region by exchanging their configuration (instance to VLAN mapping), name, and revision-level. Therefore, if you need to have two bridges in the same region, the two bridges must have identical configurations, names, and revision-levels.

Configuring MSTP

To configure a device for MSTP for 1 or more VLANs that have the same Layer 2 topology, you could configure the name and the revision on each device that is being configured for MSTP. This name is unique to each device. You must then create an MSTP Instance and assign an ID. VLANs are then assigned to MSTP instances. These instances must be configured on all devices that interoperate with the same VLAN assignments. Port cost, priority and global parameters can then be configured for individual ports and instances. In addition, operational edge ports and point-to-point links can be created and MSTP can be disabled on individual ports.

MSTP can be configured on a device with MRP. However, they are mutually exclusive on a specific VLAN. Also, MSTP can be configured on a port that is part of a LAG following the same rules as used for STP and RSTP.

Each of the commands used to configure and operate MSTP are described in the following:

Setting the MSTP name

Each device that is running MSTP is configured with a name. It applies to the device which can have many different VLANs that can belong to many different MSTP regions. By default, the name is the MAC address of the device.

To configure an MSTP name, use a command such as the following at the Global Configuration level.

```
device(config)# mstp name mstp1
```

Syntax: [no] mstp name name

The *name* parameter defines an ASCII name for the MSTP configuration. The default name is the MAC address of the device expressed as a string.

Setting the MSTP revision number

Each device that is running MSTP is configured with a revision number. It applies to the device which can have many different VLANs that can belong to many different MSTP regions.

To configure an MSTP revision number, use a command such as the following at the Global Configuration level.

```
device(config)# mstp revision 4
```

Syntax: [no] mstp revision revision-number

The *revision-number* parameter specifies the revision level for MSTP that you are configuring on the device. It can be a number from 0 and 65535.

Configuring an MSTP instance

An MSTP instance is configured with an MSTP ID for each region. Each region can contain one or more VLANs. To configure an MSTP instance and assign a range of VLANs, use a command such as the following at the Global Configuration level.

```
device(config) # mstp instance 7 vlan 4 to 7
```

Syntax: [no] mstp instance instance-number [**vlan vlan-id** | **vlan-group group-id**]

The **instance** parameter defines the number for the instance of MSTP that you are configuring. The maximum number of instances that can be configured is 16.

The **vlan** parameter assigns one or more VLANs or a range of VLANs to the instance defined in this command.

The **vlan-group** parameter assigns one or more VLAN groups to the instance defined in this command.

Configuring port priority and port path cost

Priority and path cost can be configured for a specified instance. To configure an MSTP instance, use a command such as the following at the Global Configuration level.

```
device(config)# mstp instance 7 ethernet 3/1 priority 32 path-cost 200
```

Syntax: [no] mstp instance instance-number ethernet slot/port priority port-priority path-cost cost

The *instance-number* variable is the number of the instance of MSTP that you are configuring priority and path cost for.

The **ethernet***slot/port* parameter specifies a port within a VLAN. The priority and path cost configured with this command will apply to VLAN that the port is contained within.

You can set a **priority** to the port that gives it forwarding preference over lower priority instances within a VLAN or on the device. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 240 in increments of 16. The default value is 128.

A **path-cost** can be assigned to a port to bias traffic towards or away from a path during periods of rerouting. Possible values are 1 - 200000000.

Configuring bridge priority for an MSTP instance

Priority can be configured for a specified instance. To configure priority for an MSTP instance, use a command such as the following at the Global Configuration level.

```
device(config)# mstp instance 1 priority 8192
```

Syntax: [no] mstp instance instance-number priority priority-value

The *instance-number* variable is the number for the instance of MSTP that you are configuring.

You can set a **priority** to the instance that gives it forwarding preference over lower priority instances within a VLAN or on the device. A higher number for the priority variable means a lower forwarding priority. Acceptable values are 0 - 61440 in increments of 4096. The default value is 32768.

Setting the MSTP global parameters

MSTP has many of the options available in RSTP as well as some unique options. To configure MSTP Global parameters for all instances on a device.

```
device(config)# mstp force-version 0 forward-delay 10 hello-time 4 max-age 12 max-hops 9
```

Syntax: [no] mstp force-version mode-number forward-delay value hello-time value max-age value max-hops value

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following *mode-number* values:

- 0 - The STP compatibility mode. Only STP BPDUs will be sent. This is equivalent to single STP.
- 2 - The RSTP compatibility mode. Only RSTP BPDUS will be sent. This is equivalent to single STP.
- 3 - MSTP mode. In this default mode, only MSTP BPDUS will be sent.

The **forward-delay** *value* specifies how long a port waits before it forwards an RST BPDU after a topology change. This can be a value from 4 - 30 seconds. The default is 15 seconds.

The **hello-time** *value* parameter specifies the interval between two hello packets. The parameter can have a value from 1 - 10 seconds. The default is 2 seconds.

The **max-age** *value* parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. You can specify a value from 6 - 40 seconds. The default value is 20 seconds.

The **max-hops** *value* parameter specifies the maximum hop count. You can specify a value from 1 - 40 hops. The default value is 20 hops.

Setting ports to be operational edge ports

You can define specific ports as edge ports for the region in which they are configured to connect to devices (such as a host) that are not running STP, RSTP, or MSTP. If a port is connected to an end device such as a PC, the port can be configured as an edge port. To configure ports as operational edge ports enter a command such as the following.

```
device(config)# mstp admin-edge-port ethernet 3/1
```

Syntax: [no] mstp admin-edge-port ethernet slot/port

The *slot/port* parameter specifies a port or range of ports as edge ports in the instance they are configured in.

Setting point-to-point link

You can set a point-to-point link between ports to increase the speed of convergence. To create a point-to-point link between ports, use a command such as the following at the Global Configuration level.

```
device(config)# mstp admin-pt2pt-mac ethernet 2/5 ethernet 4/5
```

Syntax: [no] mstp admin-pt2pt-mac ethernet slot/port

The *slot/port* parameter specifies a port or range of ports to be configured for point-to-point links to increase the speed of convergence.

Disabling MSTP on a port

To disable MSTP on a specific port, use a command such as the following at the Global Configuration level.

```
device(config)# mstp disable 2/1
```

Syntax: [no] mstp disable slot/port

The *slot/port* variable specifies the location of the port that you want to disable MSTP for.

Forcing ports to transmit an MSTP BPDU

To force a port to transmit an MSTP BPDU, use a command such as the following at the Global Configuration level.

```
device(config)# mstp force-migration-check ethernet 3/1
```

Syntax: [no] mstp force-migration-check ethernet slot/port

The *slot/port* variable specifies the port or ports that you want to transmit an MSTP BPDU from.

Enabling MSTP on a device

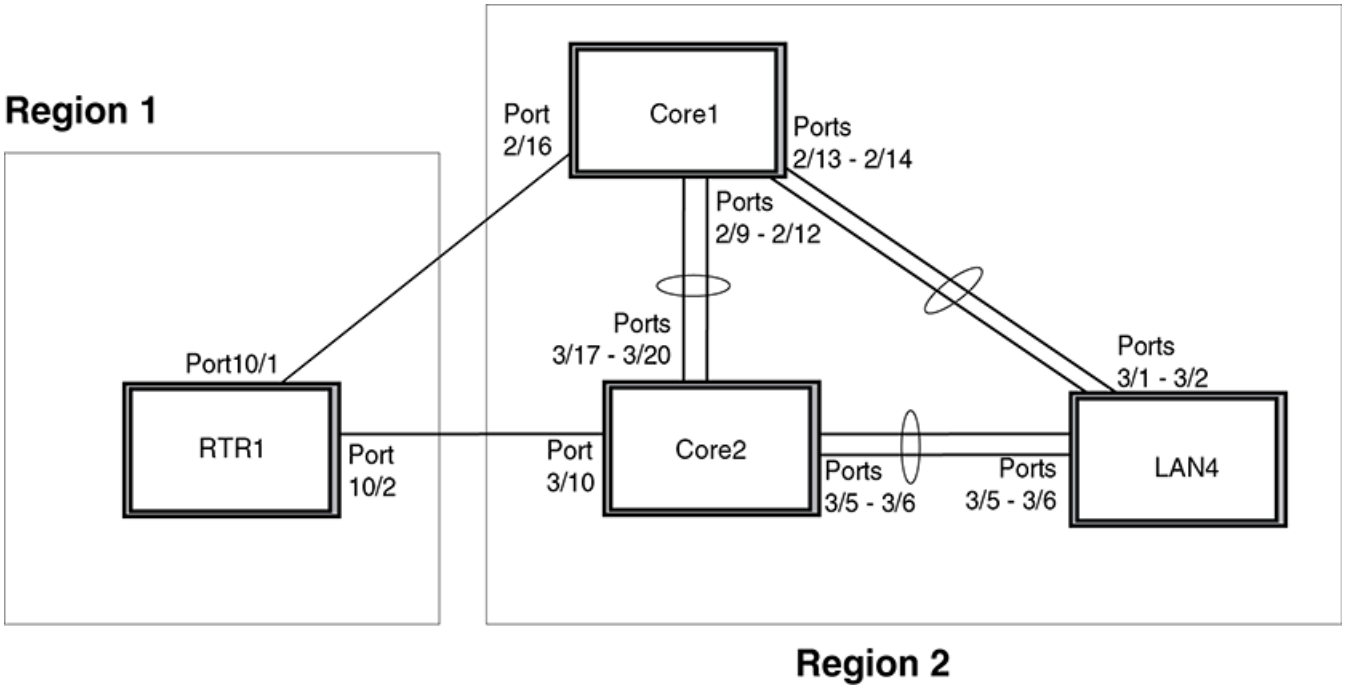
To enable MSTP on your device, use a command such as the following at the Global Configuration level.

```
device(config)# mstp start
```

Syntax: [no] start

In Figure 75 four Brocade devices are configured in two regions. There are four VLANs in four instances in Region 2. Region 1 is in the CIST.

FIGURE 73 SAMPLE MSTP configuration



RTR1 configuration

```
device(config-vlan-4093)tagged ethernet 10/1 to 10/2
device(config-vlan-4093)exit
device(config) mstp name Reg1
device(config) mstp revision 1
device(config) mstp instance 0 vlan 4093
device(config) mstp admin-pt2pt-mac ethernet 10/1 to 10/2
device(config) mstp start
device(config) hostname RTR1
```

Core 1 configuration

```
device(config-vlan-1) name DEFAULT-VLAN
device(config-vlan-1)no spanning-tree
device(config-vlan-1) exit
device(config) vlan 20
device(config-vlan-20) tagged ethernet 2/9 to 2/14 ethernet 2/16
device(config-vlan-20) no spanning-tree
device(config-vlan-20) exit
device(config) vlan 21
device(config-vlan-21) tagged ethernet 2/9 to 2/14 ethernet 2/16
device(config-vlan-21) no spanning-tree
device(config-vlan-21) exit
device(config) vlan 22
device(config-vlan-22) tagged ethernet 2/9 to 2/14 ethernet 2/16
device(config-vlan-22) no spanning-tree
device(config-vlan-22) exit
device(config) vlan 23
device(config) mstp name HR
device(config) mstp revision 2
device(config) mstp instance 20 vlan 20
device(config) mstp instance 21 vlan 21
device(config) mstp instance 22 vlan 22
device(config) mstp instance 0 priority 8192
device(config) mstp admin-pt2pt-mac ethernet 2/9 to 2/14
device(config) mstp admin-pt2pt-mac ethernet 2/16
device(config) mstp disable ethernet 2/240.
device(config) mstp start
device(config) hostname CORE1
```

Core2 configuration

```
device(config) vlan 1 name DEFAULT-VLAN
device(config-vlan-1) no spanning-tree
device(config-vlan-1) exit
device(config) vlan 20
device(config-vlan-20) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
device(config-vlan-20) no spanning-tree
device(config-vlan-20) exit
device(config) vlan 21
device(config-vlan-21) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
device(config-vlan-21) no spanning-tree
device(config-vlan-21) exit
device(config) vlan 22
device(config-vlan-22) tagged ethernet 3/5 to 3/6 ethernet 3/17 to 3/20
device(config-vlan-22) no spanning-tree
device(config-vlan-22) exit
device(config) mstp name HR
device(config) mstp revision 2
device(config) mstp instance 20 vlan 20
device(config) mstp instance 21 vlan 21
device(config) mstp instance 22 vlan 22
device(config) mstp admin-pt2pt-mac ethernet 3/17 to 3/20 ethernet 3/5 to 3/6
device(config) mstp admin-pt2pt-mac ethernet 3/10
device(config) mstp disable ethernet 3/7 ethernet 3/24
device(config) mstp start
device(config) hostname CORE2
```

LAN 4 configuration

```
device(config) vlan 1 name DEFAULT-VLAN
device(config-vlan-1) no spanning-tree
device(config-vlan-1) exit
device(config) vlan 20
device(config-vlan-20) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
device(config-vlan-20) no spanning-tree
device(config) exit
device(config) vlan 21
device(config-vlan-21) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
device(config-vlan-21) no spanning-tree
```

```
device(config-vlan-21) exit
device(config) vlan 22
device(config-vlan-22) tagged ethernet 3/1 to 3/2 ethernet 3/5 to 3/6
device(config-vlan-22) no spanning-tree
device(config) mstp config name HR
device(config) mstp revision 2
device(config) mstp instance 20 vlan 20
device(config) mstp instance 21 vlan 21
device(config) mstp instance 22 vlan 22
device(config) mstp admin-pt2pt-mac ethernet 3/5 to 3/6 ethernet 3/1 to 3/2
device(config) mstp start
device(config) hostname LAN4
```

Displaying MSTP statistics

MSTP statistics can be displayed using the commands shown below.

To display all general MSTP information, enter the following command.

```
device(config)#show mstp
MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20
Port  Pri PortPath  Role       State      Designa-  Designated
Num       Cost                          ted cost  bridge
3/1   128 2000      MASTER     FORWARDING 0         8001000cdb80af01
```

To display all general MSTP information for blocked ports only, enter the following command

```
Brocade# show mstp blocked
MSTP Instance 0 (CIST) - VLAN Scope: None
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
80000024389e2d00 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
80000024388f6b00 0         80000024388f6b00 2000      80000024388f6b00 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/2   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/3   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/4   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/5   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/6   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/7   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
3/8   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
MSTP Instance 1 - VLANs: 10
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
```

```
hex              cnt hex                       hex                    cnt
80010024389e2d00 20  80010024388f6b00 2000     80010024388f6b00 3/1   19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/2   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
3/3   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
3/4   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
MSTP Instance 2 - VLANs: 20
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
80020024389e2d00 20  80020024388f6b00 2000     80020024388f6b00 3/5   19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/6   128 2000      F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
3/7   128 2000      F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
3/8   128 2000      F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
```

The following example displays MSTP information for a specified MSTP instance.

```
device(config)# show mstp 1

MSTP Instance 1 - VLANs: 2
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001000cdb80af01 20  8001000cdb80af01 0         8001000cdb80af01 Root  20
Port  Pri PortPath  Role       State       Designa-  Designated
Num       Cost                           ted cost  bridge
3/1   128 2000      MASTER     FORWARDING  0         8001000cdb80af01
```

The following example displays blocked ports only, for a specified MSTP instance

```
Brocade# show mstp blocked 1
MSTP Instance 1 - VLANs: 10
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
80010024389e2d00 20  80010024388f6b00 2000     80010024388f6b00 3/1   19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/2   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
3/3   128 2000      F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
```

3/4 128 2000 F F ALTERNATE DISCARDING 0 80010024388f6b00

.

Refer to Table 58 for details about the display parameters.

Syntax: show mstp [**blocked**] [**mstp-id**]

The **blocked** parameter displays information for blocked ports only. When the blocked parameter is not specified, information is displayed for all ports.

The *mstp-id* variable specifies the MSTP instance for which you want to display information.

TABLE 58 Output from show MSTP

This field...	Displays...
MSTP Instance	The ID of the MSTP instance whose statistics are being displayed. For the CIST, this number is 0.
VLANs:	The number of VLANs that are included in this instance of MSTP. For the CIST this number will always be 1.

TABLE 58 Output from show MSTP (continued)

This field...	Displays...
Bridge Identifier	The MAC address of the bridge.
Bridge MaxAge sec	Displays configured Max Age.
Bridge Hello sec	Displays configured Hello variable.
Bridge FwdDly sec	Displays configured FwdDly variable.
Bridge Hop cnt	Displays configured Max Hop count variable.
Root MaxAge sec	Max Age configured on the root bridge.
Root Hello sec	Hello interval configured on the root bridge.
Root FwdDly sec	FwdDly interval configured on the root bridge.
Root Hop Cnt	Current hop count from the root bridge.
Root Bridge	Bridge identifier of the root bridge.
ExtPath Cost	The configured path cost on a link connected to this port to an external MSTP region.
Regional Root Bridge	The Regional Root Bridge is the MAC address of the Root Bridge for the local region.
IntPath Cost	The configured path cost on a link connected to this port within the internal MSTP region.
Designated Bridge	The MAC address of the bridge that sent the best BPDU that was received on this port.
Root Port	Port indicating shortest path to root. Set to "Root" if this bridge is the root bridge.
Port Num	The port number of the interface.
Pri	The configured priority of the port. The default is 128.
PortPath Cost	Configured or auto detected path cost for port.
P2P Mac	Indicates if the port is configured with a point-to-point link: • **T** - The port is configured in a point-to-point link • **F** - The port is not configured in a point-to-point link
Edge	Indicates if the port is configured as an operational edge port: • **T** - indicates that the port is defined as an edge port. • **F** - indicates that the port is not defined as an edge port
Role	The current role of the port: • Master • Root • Designated • Alternate • Backup • Disabled
State	The port's current 802.1w state. A port can have one of the following states: • Forwarding • Discarding • Learning • Disabled
Designated Cost	Port path cost to the root bridge.
Max Hop cnt	The maximum hop count configured for this instance.
Root Hop cnt	Hop count from the root bridge.

Displaying MSTP information for CIST instance 0

Instance 0 is the Common and Internal Spanning Tree Instance (CIST). When you display information for this instance there are some differences with displaying other instances. The following example displays MSTP information for CIST Instance 0.

```
device(config)#show mstp 0
MSTP Instance 0 (CIST) - VLANs: 1
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000000cdb80af01 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000000480bb9876 2000      8000000cdb80af01 0         8000000480bb9876 3/1
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
3/1   128 2000      T   F    ROOT       FORWARDING 0         8000000480bb9876
```

Refer to Displaying MSTP statistics on page 368 for explanation about the parameters in the output.

To display MSTP configuration information, enter the following command.

```
device(config)# show mstp configuration
mstp name test
mstp revision 1
mstp instance 1  vlan 100
mstp admin-pt2pt-mac ethe 4/7 to 4/8
mstp start
```

To display details about the MSTP that is configured on the device, enter the following command.

```
device(config)# show mstp detail
MSTP Instance 0 (CIST) - VLAN Scope: None
--------------------------------------------------------------------------------
  Bridge: 8000002438a5a800 [Priority 32768, SysId 0, Mac 002438a5a800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 4/7 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128, OperEdge F, OperPt2PtMac T, Boundary F
  Designated   - Root 8000002438a5a800, RegionalRoot 8000002438a5a800,
                 Bridge 8000002438a5a800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2 recover timer 0
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 5, RST 0, Config 0, TCN 0
                 Sent MST 186, RST 0, Config 0, TCN 0
Port 4/8 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128, OperEdge F, OperPt2PtMac T, Boundary F
  Designated   - Root 8000002438a5a800, RegionalRoot 8000002438a5a800,
                 Bridge 8000002438a5a800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2 recover timer 0
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 17, RST 0, Config 0, TCN 0
                 Sent MST 173, RST 0, Config 0, TCN 0
MSTP Instance 1 - VLANs: 100
--------------------------------------------------------------------------------
  Bridge: 8001002438a5a800 [Priority 32768, SysId 1, Mac 002438a5a800]
Port 4/7 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128
  Designated   - RegionalRoot 8001002438a5a800, IntCost 0
                 Bridge 8001002438a5a800
  ActiveTimers - recover timer 0
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 4/8 - Role: DESIGNATED - State: FORWARDING
  PathCost 2000, Priority 128
  Designated   - RegionalRoot 8001002438a5a800, IntCost 0
                 Bridge 8001002438a5a800
```

```
ActiveTimers - recover timer 0
MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp [**mstp-id** | **configuration** | **detail**] [**begin string** | **exclude string** | **include string**]

The *mstp-id* variable specifies the MSTP instance for which you want to display information.

The **begin***string* parameter specifies the display of information from the first line containing the "string". Information prior to the first occurence of the "string" will not be displayed.

The **exclude***string* parameter specifies the exclusion of lines containing the "string". All other information will be displayed.

The **include***string* parameter specifies the display of information containing the "string" only. All other information will be filtered out.

Interoperability between MSTP and Single STP or Single RSTP

- MSTP can interoperate with SSTP or a RSTP. However it is recommended to assign all VLANs to CIST (MSTP instance 0) while operating in a SSTP or a RSTP domain since only CIST participates in the convergence with SSTP or a RSTP.
- If the STP or RSTP is enabled on a native VLAN or on an untagged VLAN (or if SSTP or a single RSTP is configured), it sends out untagged BPDUs and can interoperate with MSTP which consumes these untagged BPDUs and converges accordingly. These untagged BPDUs can also converge with SSTP or RSTP configured on the receiving side resulting in the MSTP not being configured.
- If STP is enabled on a native VLAN or an untagged VLAN or if SSTP or RSTP is configured, it will converge with STP enabled on native or an untagged VLAN if configured, or with RSTP enabled on native or untagged VLAN if configured, or otherwise with SSTP/Single RSTP/MSTP if configured on the receiving side.

MSTP support for PBB

When applied to a PBB environment, this feature will ensure a loop free topology.

Scalability

- A maximum of 16 MSTP Instances are allowed per MSTP region.
- A maximum of 10 MSTP regions are allowed.
- A maximum of 40 MSTP Instances are allowed.

Limitations

The following restrictions apply when are using MSTP with PBB

- Dual homing of an AS configured with SVLANs in an Edge RSTP and Dual homing of a CS configured with regular VLANs in a Core MSTP is supported, but Dual homing of both AS and CS in an Edge and Core MSTP together is not supported.
- Each switch in a single MSTP region needs to be configured with unique name, but must use the same region name and revision number.
- For each MSTP region, the configured VLANs for an MSTP instance need to be of the same type, for example, either VLANs (that is, not VPLS VLANs), SVLANs, or CVLANs. This means VLANS, SVLANs, and CVLANs cannot be configured to co-exist within the same region.
- In the case of an VLAN, SVLAN, or CVLAN, the same VLAN ID cannot be configured as a part of two different MSTP instances within an MSTP region.
- MSTP configuration for VPLS VLANs with ISIDs is not supported.

- An interface can be a part of only one region when multiple regions are configured on a switch.
- Legacy and multi region MSTP configuration is not allowed at the same time on a switch.
- VPLS instances having two different VLANs is not supported.
- There is no MIB support for PBB MSTP.
- MSTP should not be configured for (1) topology groups having layer 2 (L2) member VLANs (2) member VLANs configured in a topology group. If a topology group is configured with a master vlan running MSTP, layer 2 (L2) VLANs should not be configured as members until MSTP is disabled on the master VLAN of this topology group. Such configurations via CLI are blocked.

Use case scenario

The figure below displays the use case. In this scenario, the network has two Core Switches (CSs) to provide resiliency in the core as well as service load sharing.

The CS functions as a Backbone Core Bridge (BCB). Every AS (Access Switch) in the network will dual home to the two CSs. The AS functions as a Backbone Edge Bridge (BEB). The function of the CS is to switch traffic between ASs. As a BCB, a CS will switch on the outer PBB B-tag and will not perform any PBB encapsulation.

The AS accepts 802.1q, or Q-in-Q traffic, from ESs and encapsulates the traffic into PBB frames. The SVLAN of the customer frame is the service delimiter and is mapped to a specific ISID in the PBB network. The ASs are logically interconnected with PBB tunnels (BVLANs), which always traverse a CS. The ASs will connect to the ESs as shown in the figure below.

The AS and CS switches in a network will form a single MSTP region. With the dual homing of the ASs to the CSs, all failures are protected against.

Each ES will attach to an AS and will form its own MSTP domain with the AS as the root. ES traffic that is destined for a port on the same ES does not need to enter the network core. The traffic will stay intra-switch.

Traffic that is destined for a port on another ES attached to the same AS will switch directly on the AS and will not have to enter the network core. The traffic will stay within the AS and attached ES.

Traffic that is destined for a port on an ES that is attached to a different AS will be encapsulated into PBB and will traverse the PBB core.

Figure to be added here for Ethernet Switch Architecture Use Case

Figure to be added here for AS, CS, and ES Physical Connections

Edge MSTP in a PB network

The following deployment scenario is a case where MSTP is deployed for a single S-VLAN in a PB network. PBB traffic uses S-VLAN 200.

The procedure to configure the nodes in the topology are discussed below.

Figure to be added here for Edge MSTP in a PB network.

High availability

MSTP supports MP switchover and hitless software upgrade. When an MSTP root bridge undergoes MP switchover and hitless upgrade, there will be no break in transmission of the MSTP BPDU from it during reboot of the line cards. Due to this, there will be no re-convergence of the topology and no disruption in traffic.

MSTP PBB with multi region feature also supports MP switchover and hitless software upgrade. There will be no traffic disruption during an upgrade.

MSTP PBB Configuration Commands

MSTP-region

To configure multiple MSTP regions on a single bridge to represent different bridging domains, use the **MSTP-region** command. This command is available only in the configuration mode.

All the existing MSTP commands can be executed from this mode.

```
device(config)#mstp-region 1
device(config-mstp-region-1)# mstp-region ?
```

Syntax: [no] mstp-region r *egion-id*

The acceptable range for this command is 1 to 10.

Executing the **no mstp-region** command will delete all the configurations that are configured under the **region** submode.

MSTP Instance Mapping

To configure Regular VLANs or VPLS VLANs mapped to an MSTP instance in a PB or PBB network, use the following command options.

Syntax: [no] mstp-region instance *instance-id* **vlan** *vlan-id*

Syntax: [no] mstp-region instance *instance-id* **vlan** *vlan-id* **to** *vlan-id*

Syntax: [no] mstp-region instance *instance-id* **vlan-group** *group-id*

For the Brocade NetIron MLX Series and Brocade NetIron XMR Series

On the Brocade NetIron MLX Series and Brocade NetIron XMR Series, use the following commands to configure VPLS VLANs mapped to an MSTP instance.

Syntax: [no] mstp-region instance *instance-id* **vpls** *vpls-id* **vlan** *vlan-id*

Syntax: [no] mstp-region instance *instance-id* **vpls** *vpls-id* **vlan** *vlan-id* **to** *vlan-id*

For the Brocade NetIron CER Series and Brocade NetIron CES Series

On the Brocade NetIron CER Series and Brocade NetIron CES Series, use the following command to configure VPLS VLANs mapped to an MSTP instance.

Syntax: [no] mstp-region instance *instance-id* **esi** *esi-name* **vlan** *vlan-id*

Each variable has the following range.

- Instance ID: 0-4094 (0 for CIST and 1-4094 for MSTI)
- Path Cost: 1-200000000
- Port Priority: 0-240 in the increments of 16
- Instance Priority Value: 0-61440 in the increments of 4096
- VLAN ID: 1-4090
- VPLS ID: 1-4294967294

Executing the **no mstp-region instance** command deletes the configured MST instance to VLAN mapping.

Configuring the Brocade NetIron MLX Series and Brocade NetIron XMR Series

The following procedure describes how to configure AS-1, AS-2, and ES-1 in the scenario shown in the figure above.

Configuring AS-1

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/2
```

MSTP Configuration

```
device_AS-1(config)#mstp-region 2
device_AS-1(config-mstp-region-2)#mstp-region name PB-Domain1
device_AS-1(config-mstp-region-2)#mstp-region rev 1
device_AS-1(config-mstp-region-2)#mstp-region instance 1 vpls 1 vlan 200
Brocade_AS-1(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_AS-1(config-mstp-region-2)#mstp-region start
```

Configuring AS-2

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/3
```

MSTP Configuration

```
device_AS-2(config)#mstp-region 2
device_AS-2(config-mstp-region-2)#mstp-region name PB-Domain1
device_AS-2(config-mstp-region-2)#mstp-region rev 1
device_AS-2(config-mstp-region-2)#mstp-region instance 1 vpls 1 vlan 200
Brocade_AS-2(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/3
device_AS-2(config-mstp-region-2)#mstp-region start
```

Configuring ES-1

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration:

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-svlan 1
device_ES-1(config-mpls-vpls-pb-svlan)#pbb
device_ES-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 300 on customer port.

```
device_ES-1(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-svlan-vlan-300)#tag eth 1/1 eth 1/4
```

MSTP Configuration

```
device_ES-1(config)#mstp-region 2
device_ES-1(config-mstp-region-2)#mstp-region name PB-Domain1
device_ES-1(config-mstp-region-2)#mstp-region rev 1
device_ES-1(config-mstp-region-2)#mstp-region instance 1 vpls 1 vlan 200
Brocade_ES-1(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/2 to 1/3
device_ES-1(config-mstp-region-2)#mstp-region start
device_ES-1(config)#mstp-region 100
device_ES-1(config-mstp-region-100)#mstp-region name Cust-Domain
device_ES-1(config-mstp-region-100)#mstp-region rev 1
device_ES-1(config-mstp-region-100)#mstp-region instance 1 vlan 300
Brocade_ES-1(config-mstp-region-100)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/4
device_ES-1(config-mstp-region-100)#mstp-region start
```

Configuring CE-1 and CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 300 (C-VLAN) and add port 1/1(CE-1) and 1/4 (CE-2) to it.

C-VLAN Configuration

```
device_CE-1(config)#vlan 300
device_CE-1(config-vlan-300)#tagged ethernet 1/1
```

MSTP Configuration

```
device_CE-1(config)#mstp name Cust-Domain
device_CE-1(config)#mstp rev 1
device_CE-1(config)#mstp instance 1 vlan 300
device-1(config)#mstp admin-pt2pt-mac ethernet 1/1
device_CE-1(config)#mstp start
```

C-VLAN Configuration

```
device_CE-2(config)#vlan 300
device_CE-2(config-vlan-300)#tagged ethernet 1/4
```

MSTP Configuration

```
device_CE-2(config)#mstp name Cust-Domain
device_CE-2(config)#mstp rev 1
device_CE-2(config)#mstp instance 1 vlan 300
Brocade_CE-2(config)#mstp admin-pt2pt-mac ethernet 1/4
device_CE-2(config)#mstp start
```

CES/CER configuration

Configuring AS1

```
device_AS1(config)#int e 1/1
device_AS1(config-if-e1000-1/1)#port-type backbone-edge
device_AS1(config-if-e1000-1/1)#int e 1/2
device_AS1(config-if-e1000-1/2)#port-type backbone-edge
device_AS1(config-if-e1000-1/2)#esi svlan1 encap svlan
device_AS1(config-esi-svlan1)#vlan 200
device_AS1(config-esi-svlan1-vlan-200)#tag e 1/1 e 1/2
```

Configuring AS2

```
device_AS2(config)#int e 1/1
device_AS2(config-if-e1000-1/1)#port-type backbone-edge
device_AS2(config-if-e1000-1/1)#int e 1/3
device_AS2(config-if-e1000-1/3)#port-type backbone-edge
device_AS2(config-if-e1000-1/3)#esi svlan1 encap svlan
device_AS2(config-esi-svlan1)#vlan 200
device_AS2(config-esi-svlan1-vlan-200)#tag e 1/1 e 1/3
```

MSTP configuration for AS1 and AS2

```
device_AS2(config)#mstp-region 2
device_AS2(config-mstp-region-2)#mstp-region name PB-Domain1
device_AS2(config-mstp-region-2)#mstp-region rev 1
device_AS2(config-mstp-region-2)#mstp-region instance 1 esi svlan1 vlan 200
Brocade_AS2(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/3
device_AS2(config-mstp-region-2)#mstp-region start
```

Configuring ES1

```
device_ES1(config)#int e 1/1
device_ES1(config-if-e1000-1/1)#port-type customer-edge
device_ES1(config-if-e1000-1/1)#int e 1/4
device_ES1(config-if-e1000-1/4)#port-type customer-edge
device_ES1(config-if-e1000-1/4)#esi cvlan1 encap cvlan
device_ES1(config-esi-cvlan1)#vlan 300
device_ES1(config-esi-cvlan1-vlan-300)#tag e 1/1 e 1/4
device_ES1(config-esi-cvlan1-vlan-300)#int e 1/2
device_ES1(config-if-e1000-1/2)#port-type provider-network
device_ES1(config-if-e1000-1/2)#int e 1/3
device_ES1(config-if-e1000-1/3)#port-type provider-network
device_ES1(config-if-e1000-1/3)#esi svlan2 encap svlan
device_ES1(config-esi-svlan2)#esi-client cvlan1
device_ES1(config-esi-svlan2)#vlan 200
device_ES1(config-esi-svlan2-vlan-200)#tag e 1/2 e 1/3
```

MSTP configuration for ES1

```
device_ES1(config)#mstp-region 2
device_ES1(config-mstp-region-2)#mstp-region name PB-Domain1
```

```
device_ES1(config-mstp-region-2)#mstp-region rev 1
device_ES1(config-mstp-region-2)#mstp-region instance 1 esi svlan2 vlan 200
Brocade_ES1(config-mstp-region-2)#mstp-region admin-pt2pt-mac ethernet 1/2 to 1/3
device_ES1(config-mstp-region-2)#mstp-region start
device_ES1(config)#mstp-region 100
device_ES1(config-mstp-region-100)#mstp-region name Cust-Domain
device_ES1(config-mstp-region-100)#mstp-region rev 1
device_ES1(config-mstp-region-100)#mstp-region instance 1 vlan 300
Brocade_ES1(config-mstp-region-100)#mstp-region admin-pt2pt-mac ethernet 1/1 ethernet 1/4
device_ES1(config-mstp-region-100)#mstp-region start
```

Configuring CE1

```
device_CE1(config)#vlan 300
device_CE1(config-vlan-300)#tag e 1/1
```

MSTP configuration for CE1

```
device_CE1(config)#mstp name Cust-Domain
device_CE1(config)#mstp rev 1
device_CE1(config)#mstp instance 1 vlan 300
Brocade_CE1(config)#mstp admin-pt2pt-mac ethernet 1/1
device_CE1(config)#mstp start
```

Configuring CE2

```
device_CE2(config)#vlan 300
device_CE2(config-vlan-300)#tag e 1/4
```

MSTP configuration for CE2

```
device_CE2(config)#mstp name Cust-Domain
device_CE2(config)#mstp rev 1
device_CE2(config)#mstp instance 1 vlan 300
Brocade_CE-2(config)#mstp admin-pt2pt-mac ethernet 1/4
device_CE2(config)#mstp start
```

Configuring MSTP in a PBB network

The following deployment scenario is a case where MSTP is deployed for a single B-VLAN in a PBB network. PBB traffic uses B-VLAN 100. The procedure to configure the nodes in the topology are discussed below.

Figure to be added here for Core MSTP in a PBB network.

Brocade NetIron XMR Series and Brocade NetIron MLX Series configuration

AS-1 Configuration

NOTE

The configuration of AS-2 is similar to the AS-1 configuration.

For PBB traffic you will configure a VPLS instance, the B-VLAN used here is 100. For PB traffic, the S-VLAN used is 200 and C-VLAN 300. Here, you need topology groups to configure the BVLAN as the master VLAN and VPLS VLANs with different ISIDs mapping to the same BVLAN as member VLANs. Then the MSTP instance is configured to be mapped to the master VLAN of the topology group (the BVLAN).

Tag type configuration

```
device_AS-1(config)#tag-type 88e8 eth 1/1
device_AS-1(config)#tag-type 88e8 eth 1/2
device_AS-1(config)#tag-type 9100 eth 1/10
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 100
device_AS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan 1
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb) #vlan 100 isid 101010
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan1 2
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb) #vlan 100 isid 10101
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan2 3
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb) #vlan 100 isid 1010
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-bvlan)#vlan 200
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-200) #tagged ethernet 1/10
```

Topology Group Configuration

```
device_AS-1(config)#topology-group 1
device_AS-1(config-topo-group-1)#master-vlan 100
device_AS-1(config-topo-group-1)#member-vlan vpls name bvlan vlan 100 isid 101010
device_AS-1(config-topo-group-1)#member-vlan vpls name bvlan1 vlan 100 isid 10101
device_AS-1(config-topo-group-1)#member-vlan vpls name bvlan2 vlan 100 isid 1010
```

MSTP Configuration

```
device_AS-1(config)#mstp-region 1
device_AS-1(config-mstp-region-1)#mstp-region name PBB-Domain
device_AS-1(config-mstp-region-1)#mstp-region rev 1
device_AS-1(config-mstp-region-1)#mstp-region instance 1 vlan 100
Brocade_AS-1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_AS-1(config-mstp-region-1)#mstp-region start
```

CS-1 Configuration

NOTE

The CS-2 configuration is similar to the CS-1 configuration.

Port type configuration

```
device_CS-1(config)#tag-type 88e8 eth 1/1
device_CS-1(config)#tag-type 88e8 eth 1/2
```

B-VLAN Configuration:

```
device_CS-1(config)#vlan 100
device_CS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

MSTP Configuration

```
device_CS-1(config)#mstp-region 1
device_CS-1(config-mstp-region-1)#mstp-region name PBB-Domain
device_CS-1(config-mstp-region-1)#mstp-region rev 1
device_CS-1(config-mstp-region-1)#mstp-region instance 1 vlan 100
Brocade_CS-1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_CS-1(config-mstp-region-1)#mstp-region start
```

Configuring the Brocade NetIron CER Series and Brocade NetIron CES Series

Configuring AS1

NOTE

The AS-2 configuration is similar to the AS-1 configuration.

```
device_AS1(config)#int e 1/1
device_AS1(config-if-e1000-1/1)#port-type backbone-network
device_AS1(config-if-e1000-1/1)#int e 1/2
device_AS1(config-if-e1000-1/2)#port-type backbone-network
device_AS1(config-if-e1000-1/2)#int e 1/10
device_AS1(config-if-e1000-1/10)#port-type backbone-edge
device_AS1(config-if-e1000-1/10)#esi svlan2 encap svlan
device_AS1(config-esi-svlan2)#vlan 200
device_AS1(config-esi-svlan2-vlan-200)#tag e 1/10
device_AS1(config-esi-svlan2-vlan-200)#esi isid1 encap isid
device_AS1(config-esi-isid1)#isid 101010
device_AS1(config-esi-isid1-isid-101010)#esi-client svlan2
device_AS1(config-esi-isid1-isid-101010)#esi pbb-bvlan encap bvlan
device_AS1(config-esi-pbb-bvlan)#esi-client isid1
device_AS1(config-esi-pbb-bvlan)#vlan 100
device_AS1(config-esi-pbb-bvlan-vlan-100)#tag e 1/1 e 1/2
```

MSTP configuration for AS1

```
device_AS1(config)#mstp-region 1
device_AS1(config-mstp-region-1)#mstp name PBB-Domain
device_AS1(config-mstp-region-1)#mstp rev 1
device_AS1(config-mstp-region-1)#mstp instance 1 esi pbb-bvlan vlan 100
Brocade_AS1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_AS1(config-mstp-region-1)#mstp start
```

Configuring CSs

```
device_CS1(config)#int e 1/1
device_CS1(config-if-e1000-1/1)#port-type backbone-network
device_CS1(config-if-e1000-1/1)#int e 1/2
device_CS1(config-if-e1000-1/2)#port-type backbone-network
device_CS1(config-vlan-100)#esi pbb-bvlan encap bvlan
device_CS1(config-esi-pbb-bvlan)#vlan 100
device_CS1(config-esi-pbb-bvlan-vlan-100)#tag e 1/1 e 1/2
```

MSTP configuration:

```
device_CS1(config)#mstp-region 1
device_CS1(config-mstp-region-1)#mstp-region name PBB-Domain
device_CS1(config-mstp-region-1)#mstp-region rev 1
device_CS1(config-mstp-region-1)#mstp-region instance 1 esi pbb-bvlan vlan 100
Brocade_CS1(config-mstp-region-1)#mstp-region admin-pt2pt-mac ethernet 1/1 to 1/2
device_CS1(config-mstp-region-1)#mstp-region start
```

Show commands

The output of the following commands include the information about the configured Layer 2 VLANs, VPLS VLANs (Brocade NetIron MLX Series and Brocade NetIron XMR Series) or ESI VLANs (Brocade NetIron CER Series and Brocade NetIron CES Series) within MSTP.

Show MSTP config

The **show mstp config** command displays the MSTP configuration as it appears in the running config.

For Brocade NetIron MLX Series and Brocade NetIron XMR Series

```
device_DUT# show mstp config
Mstp-region 1
Mstp-region name PBB-Domain
Mstp-region revision 1
Mstp-region instance 1 vpls 1 vlan 100
Mstp-region start
Mstp-region 2
Mstp-region name PB-Domain1
Mstp-region revision 1
Mstp-region instance 1 vpls 1 vlan 200
Mstp-region start
```

Syntax: show mstp config

For Brocade NetIron CER Series and Brocade NetIron CES Series

```
device_DUT# show mstp config
Mstp-region 1
Mstp-region name PBB-Domain
Mstp-region revision 1
Mstp-region instance 1 esi pbb-bvlan vlan 100
Mstp-region start
Mstp-region 2
Mstp-region name PB-Domain1
Mstp-region revision 1
Mstp-region instance 1 esi pb-svlan vlan 200
Mstp-region start
```

Syntax: show mstp config

Show MSTP

The **show mstp** command displays information about all the MSTP regions for all configured instances.

For Brocade NetIron MLX Series and Brocade NetIron XMR Series

The following example displays information about all the MSTP regions for all configured instances.

```
device# show mstp
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
-------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
```

```
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
Region 2:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 200
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example displays information about blocked ports only, for all MSTP regions and all configured instances in a VPLS VLAN.

```
device# sh mstp blocked
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000748ef82ba800 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
80000024388f6b00 0         80000024388f6b00 20000     80000024388f6b00 1/13
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
MSTP Instance 1 - VPLS VLANs: 200 to 201
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001748ef82ba800 20  80010024388f6b00 20000     80010024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                        ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
MSTP Instance 2 - VPLS VLANs: 300
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
```

```
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8002748ef82ba800 20  80020024388f6b00 20000     80020024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
```

1/14 128 20000 F F ALTERNATE DISCARDING 0 80020024388f6b00

Syntax: show mstp [blocked]

For Brocade NetIron CER Series and Brocade NetIron CES Series

```
device# show mstp
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
-------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
-------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
Region 2:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
-------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - ESI VLANs: esi pb-svlan - Encapsulation svlan - vlan 200
-------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example displays MSTP information for blocked ports only, in an ESI VLAN configuration.

```
device# show mstp blocked

MSTP Instance 0 (CIST) - ESI VLAN Scope: None
--------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedb59a40 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedb4e740 20000     8000001bedb59a40 0         8000001bedb4e740 1/17
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 0         8000001bedb4e740
MSTP Instance 1 - ESI VLANs: 10 to 11
--------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedb59a40 20  8001001bedb59a40 0         8001001bedb59a40 Root  20
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 0         8001001bedb59a40
```

Syntax: show mstp [**blocked**]

Show mstp region

The **show mstp region** command displays similar output as the **show mstp** command, filtered for the queried region ID.

For Brocade NetIron MLX Series and Brocade NetIron XMR Series

The following example displays MSTP information for region "1".

```
device# show mstp region 1
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
--------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
--------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example displays MSTP information for blocked ports only, in region "1" in a VPLS VLAN configuration.

```
device#show mstp blocked region 1
Region 1
--------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000748ef82ba800 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
80000024388f6b00 0         80000024388f6b00 20000     80000024388f6b00 1/13
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80000024388f6b00
MSTP Instance 1 - VPLS VLANs: 100
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001748ef82ba800 20  80010024388f6b00 20000     80010024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80010024388f6b00
MSTP Instance 2 - VPLS VLANs: 200
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8002748ef82ba800 20  80020024388f6b00 20000     80020024388f6b00 1/13  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/14  128 20000     F   F    ALTERNATE  DISCARDING 0         80020024388f6b00
```

Syntax: show mstp [blocked] region *region-id*

For Brocade NetIron CER Series and Brocade NetIron CES Series

```
device# show mstp region 1
Region 1:
----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
--------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedaf7800 20     2      15     20     20     2     15     20
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedaf7800 0         8000001bedaf7800 0         8000001bedaf7800 Root
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8000001bedaf7800
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
--------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedaf7800 20  8001001bedaf7800 0         8001001bedaf7800 Root  20

Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
2/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
3/5   128 20000     F   F    DESIGNATED FORWARDING 0         8001001bedaf7800
```

The following example shows MSTP information for blocked ports only, filtered for region "1" in an ESI VLAN configuration.

```
device#show mstp blocked region 1
Region 1
--------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------------
Bridge           Bridge Bridge Bridge Bridge Root   Root  Root   Root
Identifier       MaxAge Hello  FwdDly Hop    MaxAge Hello FwdDly Hop
hex              sec    sec    sec    cnt    sec    sec   sec    cnt
8000001bedb59a40 20     2      15     20     20     2     15     19
Root             ExtPath   RegionalRoot     IntPath   Designated       Root
Bridge           Cost      Bridge           Cost      Bridge           Port
hex                        hex                        hex
8000001bedb4e740 0         8000001bedb4e740 20000     8000001bedb4e740 1/17
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 0         8000001bedb4e740
MSTP Instance 1 - ESI VLANs: 10
------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8001001bedb59a40 20  8001001bedb4e740 20000     8001001bedb4e740 1/17  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 20000     8001001bedb59a40
MSTP Instance 2 - ESI VLANs: 11
------------------------------------------------------------------------------
Bridge           Max RegionalRoot     IntPath   Designated       Root  Root
Identifier       Hop Bridge           Cost      Bridge           Port  Hop
hex              cnt hex                        hex                    cnt
8002001bedb59a40 20  8002001bedb4e740 20000     8002001bedb4e740 1/18  19
Port  Pri PortPath  P2P Edge Role       State      Designa-  Designated
Num       Cost      Mac Port                      ted cost  bridge
1/18  128 20000     F   F    ALTERNATE  DISCARDING 20000     8002001bedb59a40
```

Syntax: show mstp [**blocked**] **region** *region-id*

Show MSTP detail

The **show mstp detail** command displays information about all of the MSTP regions for all configured instances in detail.

For Brocade NetIron MLX Series and Brocade NetIron XMR Series

```
device_DUT# show mstp detail
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
```

```
                Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Region 2:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail

For Brocade NetIron CER Series and Brocade NetIron CES Series

```
device_DUT# show mstp detail
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
```

```
                Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Region 2:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - ESI VLANs: esi pb-svlan - Encapsulation svlan - vlan 200
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail

Show MSTP detail region

The **show mst detail region** command output is similar to the **show mstp detail** command, but is filtered for the queried region ID.

For Brocade NetIron MLX Series and Brocade NetIron XMR Series

```
device_DUT# show mstp detail region 1
Region 1:
-----------
MSTP Instance 0 (CIST) - VPLS VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - VPLS VLANs: VPLS 1 VLAN 100
------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail region *region-id*

For Brocade NetIron CER Series and Brocade NetIron CES Series

```
device_DUT# show mstp detail region 1
Region 1:
-----------
MSTP Instance 0 (CIST) - ESI VLAN Scope: None
------------------------------------------------------------------------
  Bridge: 8000001bedaf7800 [Priority 32768, SysId 0, Mac 001bedaf7800]
  FwdDelay 15, HelloTime 2, MaxHops 20, TxHoldCount 6
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
                 Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128, OperEdge F, OperPt2PtMac F, Boundary F
  Designated   - Root 8000001bedaf7800, RegionalRoot 8000001bedaf7800,
```

```
                Bridge 8000001bedaf7800, ExtCost 0, IntCost 0
  ActiveTimers - helloWhen 2
  MachineState - PRX-RECEIVE, PTX-IDLE, PPM-SENDING_RSTP, PIM-CURRENT
                 PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
  BPDUs        - Rcvd MST 811, RST 0, Config 0, TCN 0
                 Sent MST 811, RST 0, Config 0, TCN 0
MSTP Instance 1 - ESI VLANs: esi pbb-bvlan - Encapsulation bvlan - vlan 100
--------------------------------------------------------------------------------
  Bridge: 8001001bedaf7800 [Priority 32768, SysId 1, Mac 001bedaf7800]
Port 2/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
Port 3/5 - Role: DESIGNATED - State: FORWARDING
  PathCost 20000, Priority 128
  Designated   - RegionalRoot 8001001bedaf7800, IntCost 0
                 Bridge 8001001bedaf7800
  ActiveTimers -
  MachineState - PIM-CURRENT, PRT-ACTIVE_PORT, PST-FORWARDING, TCM-ACTIVE
```

Syntax: show mstp detail region *region-id*

Configuring Rapid Spanning Tree Protocol

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

> NOTE
> In addition to the features described in this chapter, refer to Root Guard and BPDU Guard in the *Configuring Spanning Tree Protocol* chapter for more details.

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

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

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

2 x FORWARD_DELAY + BRIDGE_MAX_AGE.

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

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

Bridges and bridge port roles

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

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

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

- Designated port ID

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

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

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

Ports can have one of the following roles:

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

Assignment of port roles

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

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

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

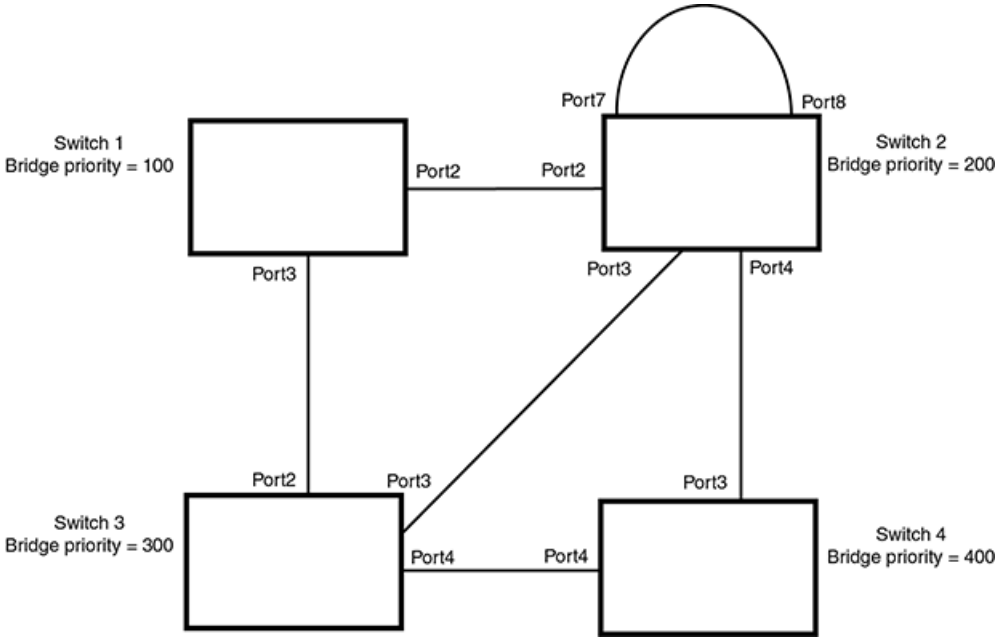
- The port that receives the RST BPDU with the lowest path cost from the root bridge becomes the Root port .
- If two ports on the same bridge are physically connected, the port that receives the superior RST BPDU becomes the Backup port , while the other port becomes the Designated port .
- If a non-root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

The following example (Figure 76) 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 76 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 74 Simple RSTP topology



Ports on Switch 1

All ports on Switch 1, the root bridge, are assigned Designated port roles.

Ports on Switch 2

Port2 on Switch 2 directly connects to the root bridge; therefore, Port2 is the Root port.

Switch 2's bridge priority value is superior to that of Switch 3 and Switch 4; therefore, the ports on Switch 2 that connect to Switch 3 and Switch 4 are given the Designated port role.

Furthermore, Port7 and Port8 on Switch 2 are physically connected. The RST BPDUs transmitted by Port7 are superior to those Port8 transmits. Therefore, Switch 2 is the Backup port and Port7 is the Designated port.

Ports on Switch 3

Port2 on Switch 3 directly connects to the Designated port on the root bridge; therefore, it assumes the Root port role.

The root path cost of the RST BPDUs received on Port4/Switch 3 is inferior to the RST BPDUs transmitted by the port; therefore, Port4/Switch 3 becomes the Designated port.

Similarly, Switch 3 has a bridge priority value inferior to Switch 2. Port3 on Switch 3 connects to Port 3 on Switch 2. This port will be given the Alternate port role, since a Root port is already established on this bridge.

Ports Switch 4

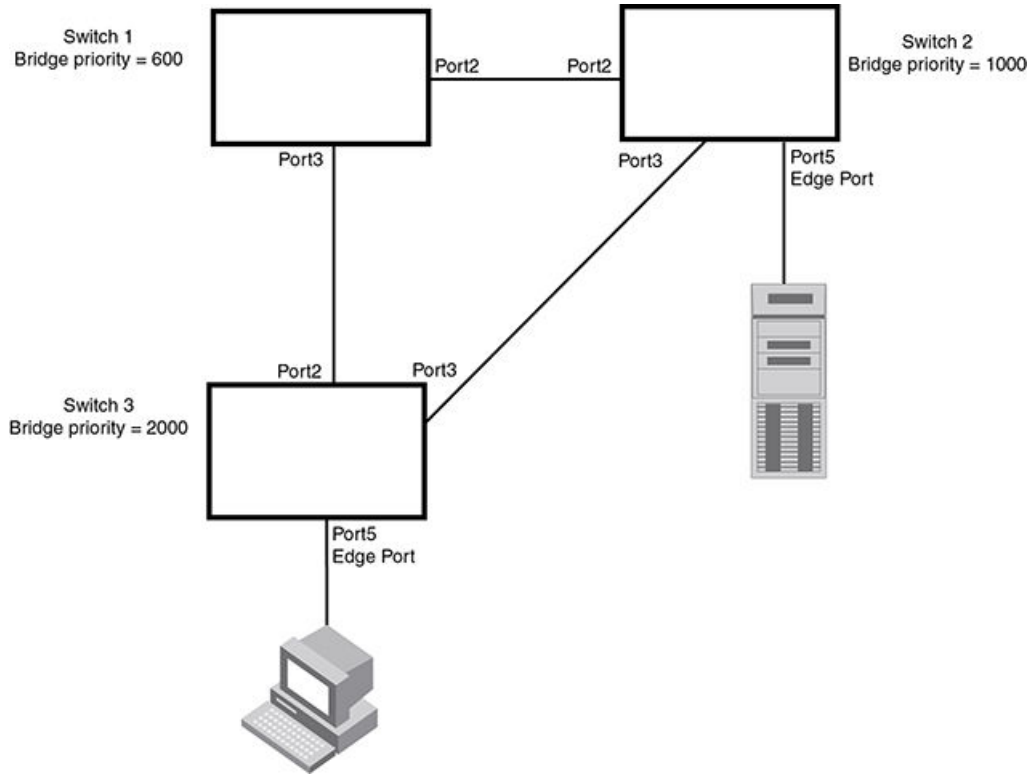
Switch 4 is not directly connected to the root bridge. It has two ports with superior incoming RST BPDUs from two separate LANs: Port3 and Port4. The RST BPDUs received on Port3 are superior to the RST BPDUs received on port 4; therefore, Port3 becomes the Root port and Port4 becomes the Alternate port.

Edge ports and Edge port roles

Brocade's implementation of RSTP allows ports that are configured as Edge ports to be present in an RSTP topology. (Figure 77). 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 75 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.

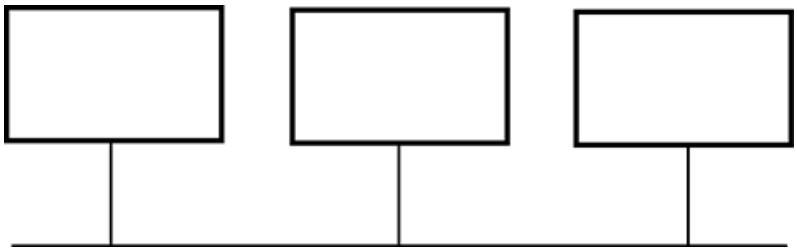
Point-to-point ports

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 78 is an example of shared media that should not be configured as point-to-point links. In Figure 78, a port on a bridge communicates or is connected to at least two ports.

FIGURE 76 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 Handshake when no root port is elected on page 397, 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

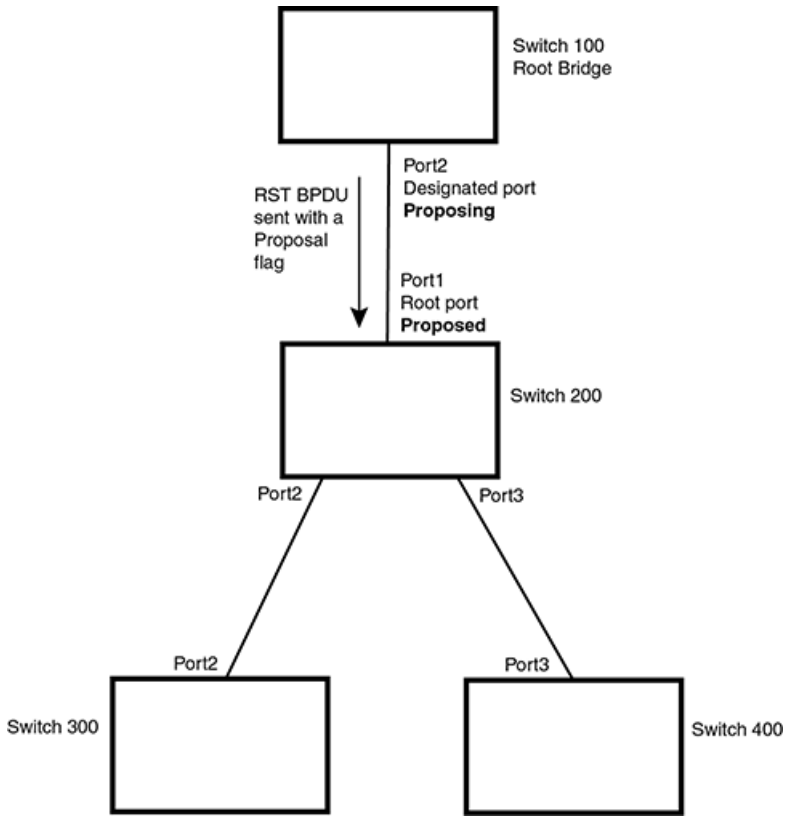
If a Root port has not been assigned on a bridge, RSTP uses the Proposing -> Proposed -> Sync -> Synced -> Agreed handshake:

- 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 (see the diagram below). The Designated port continues to send this flag in its RST BPDU until it is placed in a forwarding state (see the diagram titled “Agree stage” below) or is forced to operate in 802.1D mode. (Refer to the Compatibility of RSTP with 802.1D section)
- 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 (see the diagram below):
  - If the RST BPDU that the port receives is superior to what it can transmit, the port assumes the role of a Root port. (Refer to the section on Bridges and bridge port roles.)
  - 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.

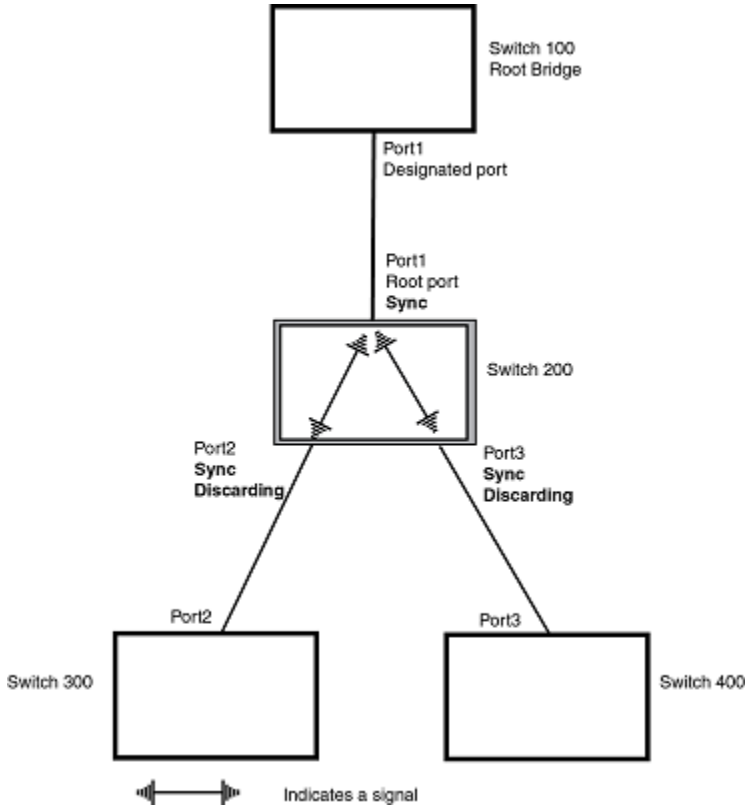
In the following diagram, Port3/Switch 200 is elected as the Root port.

FIGURE 77 Proposing and proposed stage



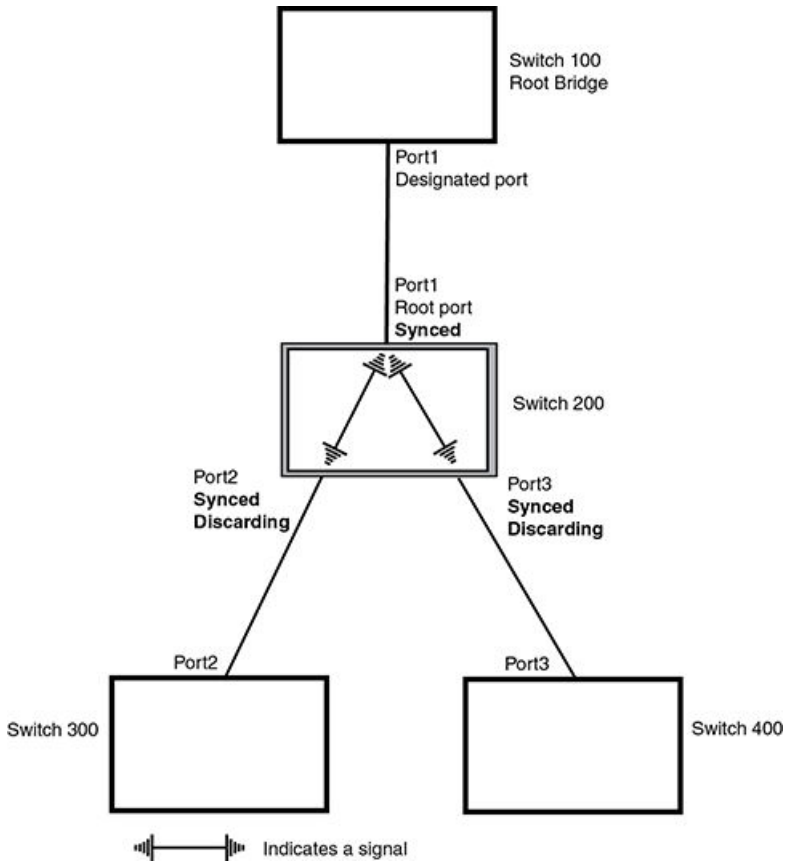
- Sync - Once the Root port is elected, it sets a sync signal on all the ports on the bridge. The signal tells the ports to synchronize their roles and states (see the following diagram). Ports that are non-edge ports with a role of Designated port change into a discarding state. These ports have to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

FIGURE 78 Sync stage



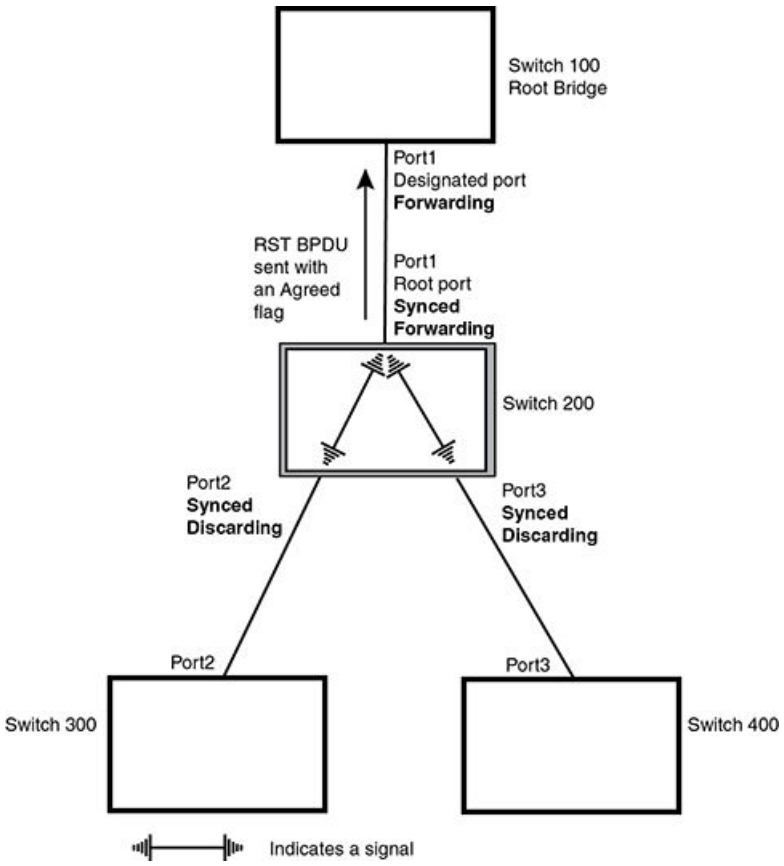
- **Synced** - Once the Designated port changes into a discarding state, it asserts a synced signal. Immediately, Alternate ports and Backup ports are synced. The Root port monitors the synced signals from all the bridge ports. Once all bridge ports asserts a synced signal, the Root port asserts its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

FIGURE 79 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 80 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.

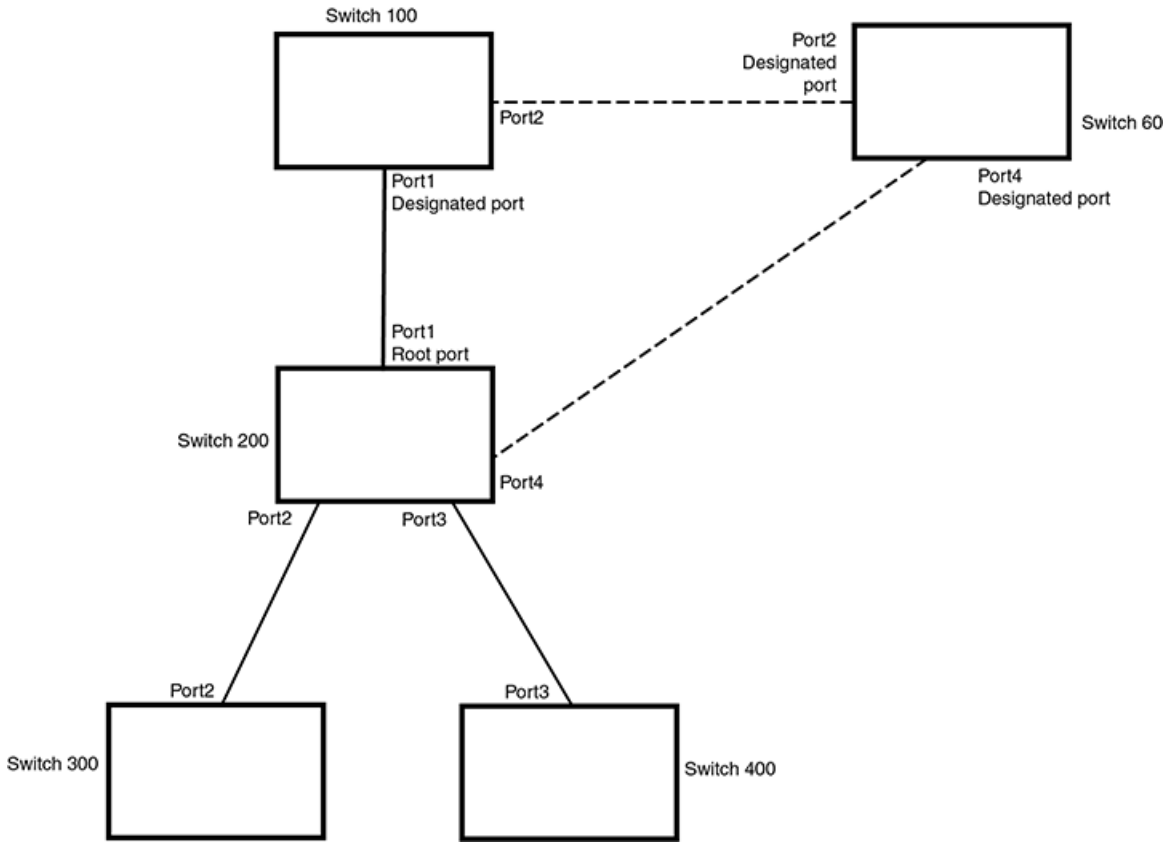
For example, Port2/Switch 200 sends an RST BPDU to Port2/Switch 300 that contains a proposal flag. Port2/Switch 300 asserts a proposed signal. Ports in Switch 300 then set sync signals on the ports to synchronize and negotiate their roles and states. Then the ports assert a synced signal and when the 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 83, a new root bridge is added to the topology.

FIGURE 81 Addition of a new root bridge

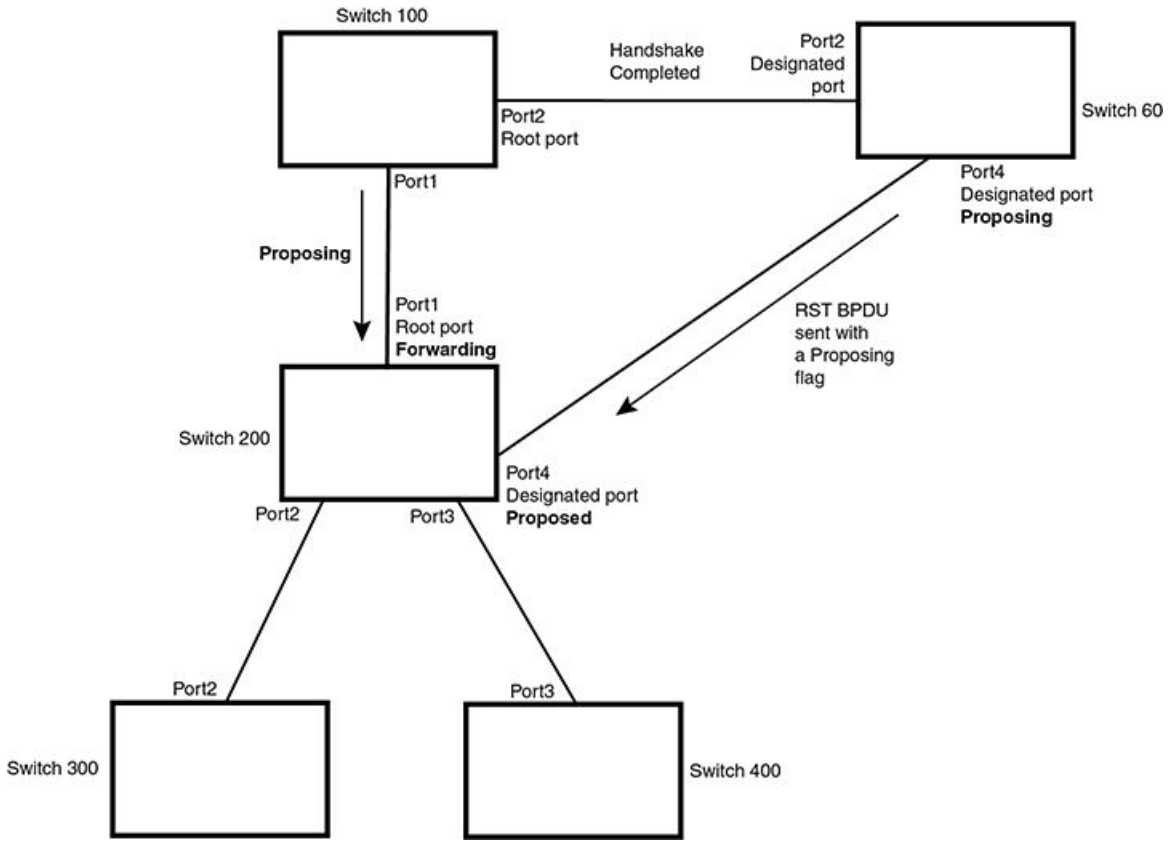


The handshake that occurs between Switch 60 and Switch 100 follows the one described in the previous section (Handshake when no root port is elected on page 397). The former root bridge becomes a non-root bridge and establishes a Root port (Figure 84).

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:

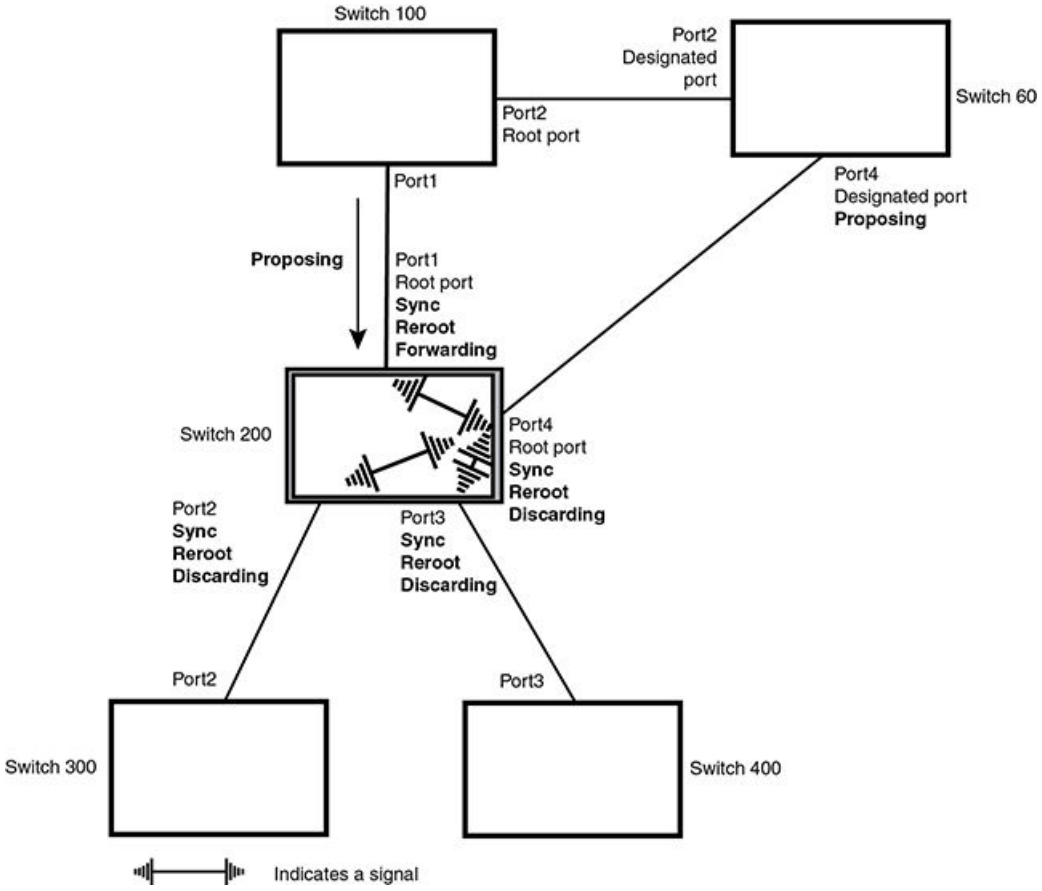
- **Proposing and Proposed** - The Designated port on the new root bridge (Port4/Switch 60) sends an RST BPDU that contains a proposing signal to Port4/Switch 200 to inform the port that it is ready to put itself in a forwarding state (Figure 84). RSTP algorithm determines that the RST BPDU that Port4/Switch 200 received is superior to what it can generate, so Port4/Switch 200 assumes a Root port role.

FIGURE 82 New root bridge sending a proposal flag



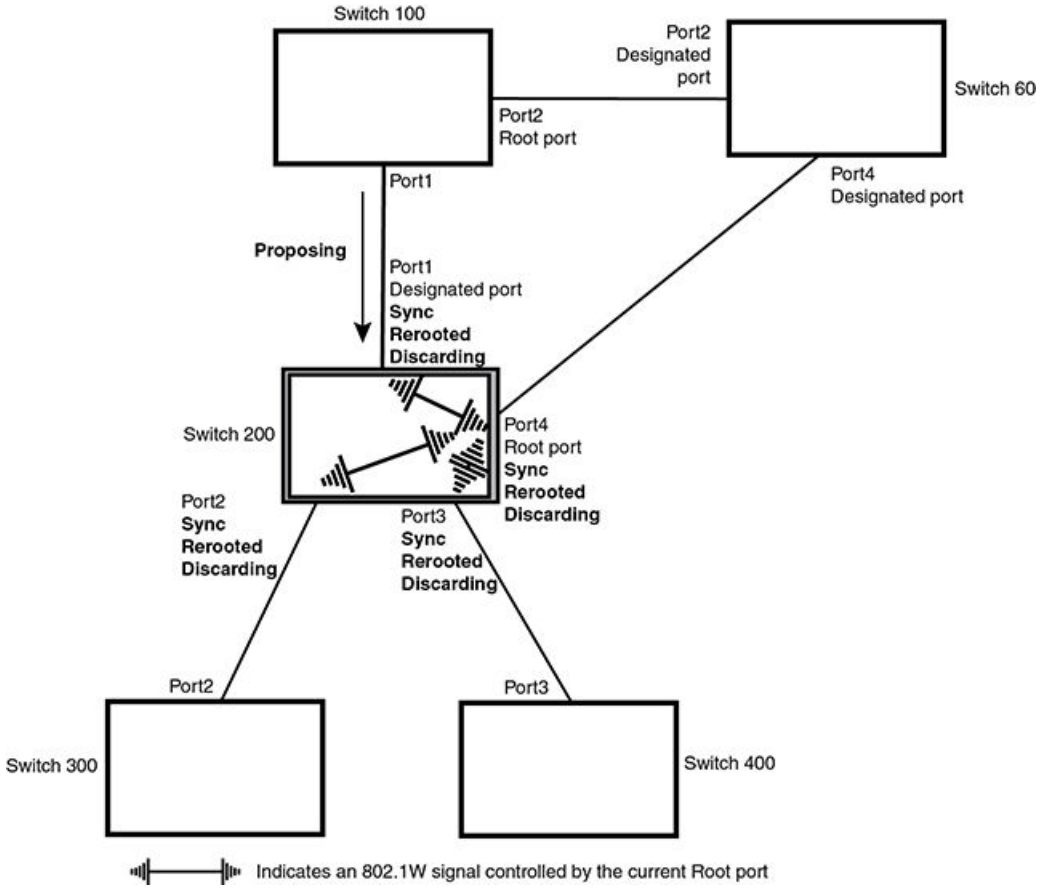
- Sync and Reroot - The Root port then asserts a sync and a reroot signal on all the ports on the bridge. The signal tells the ports that a new Root port has been assigned and they are to renegotiate their new roles and states. The other ports on the bridge assert their sync and reroot signals. Information about the old Root port is discarded from all ports. Designated ports change into discarding states (Figure 85).

FIGURE 83 Sync and reroot



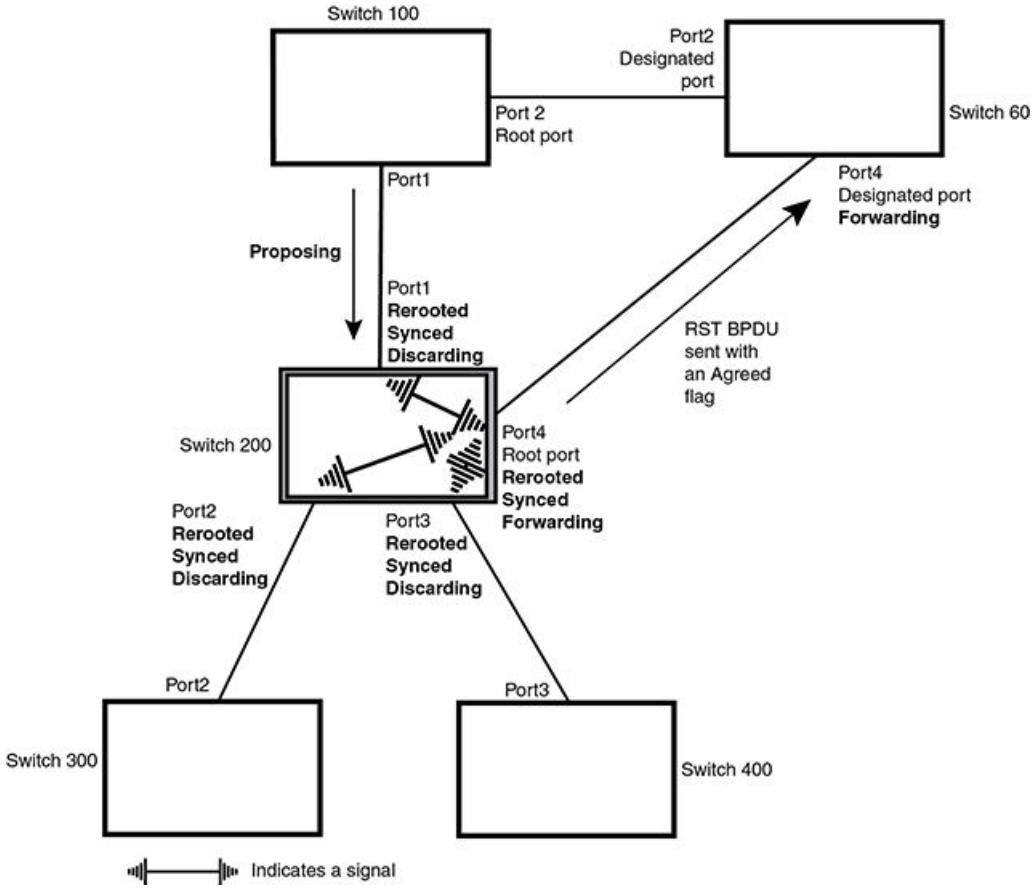
- **Sync and Rerooted** - When the ports on Switch 200 have completed the reroot phase, they assert their rerooted signals and continue to assert their sync signals as they continue in their discarding states. They also continue to negotiate their roles and states with their peer ports (Figure 86).

FIGURE 84 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 86). The Root port also moves into a forwarding state.

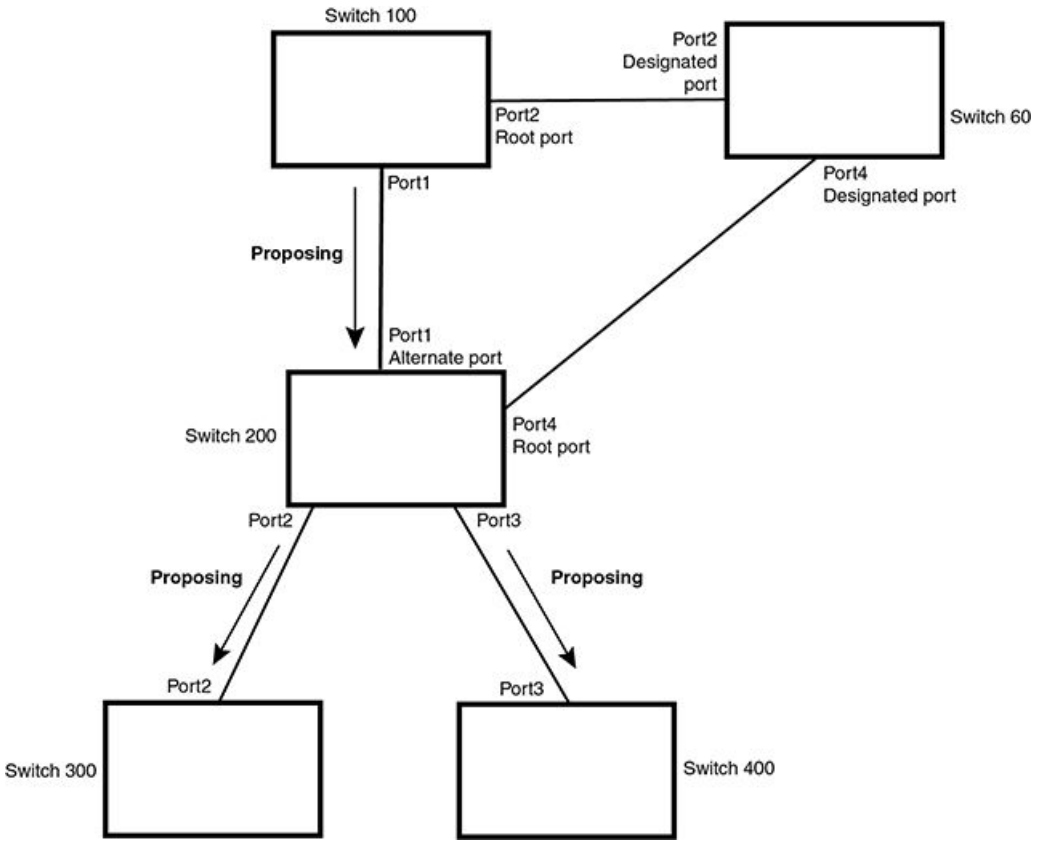
FIGURE 85 Rerooted, synced, and agreed



The old Root port on Switch 200 becomes an Alternate Port (Figure 88). 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 86 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

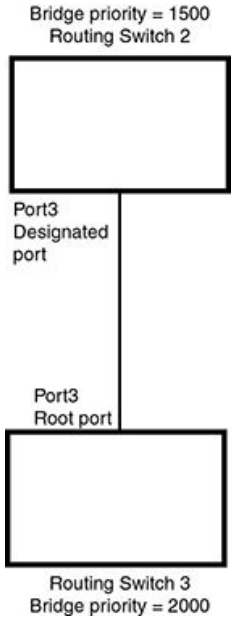
The examples in this section illustrate how RSTP convergence occurs in a simple Layer 2 topology at start-up.

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

Convergence at start up

In Figure 89, 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 87 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.

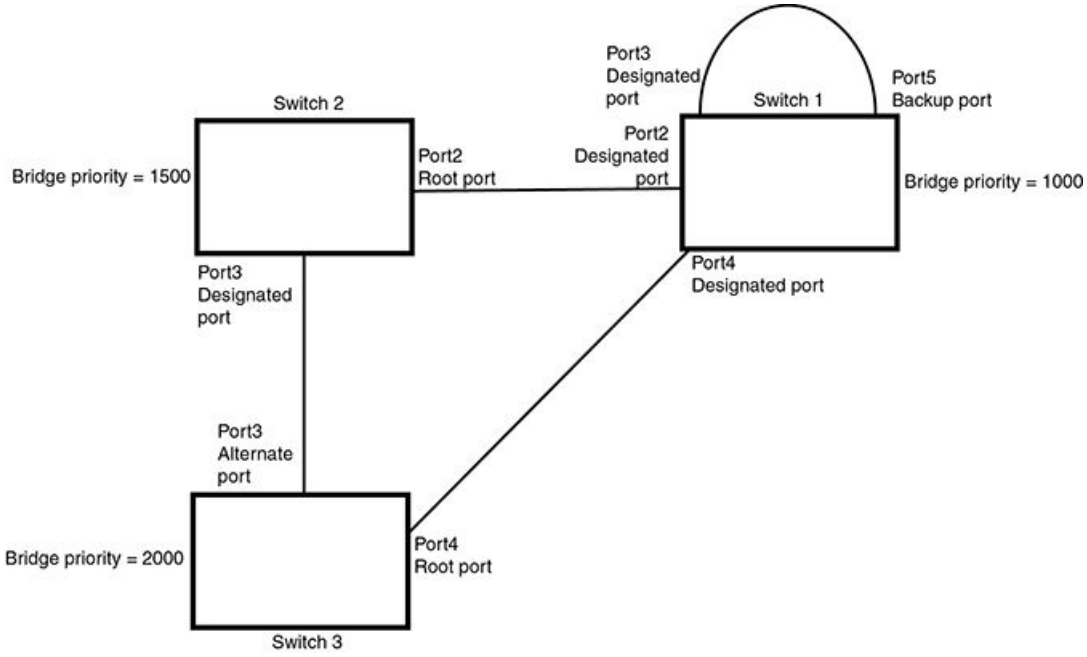
Port3/Switch 3, which starts with a role of Designated port, receives the RST BPDU and finds that it is superior to what it can transmit; therefore, Port3/Switch 3 assumes a new port role, that of a Root port. Port3/Switch 3 transmits an RST BPDU with an agreed flag back to Switch 2 and immediately goes 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 90).

FIGURE 88 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.

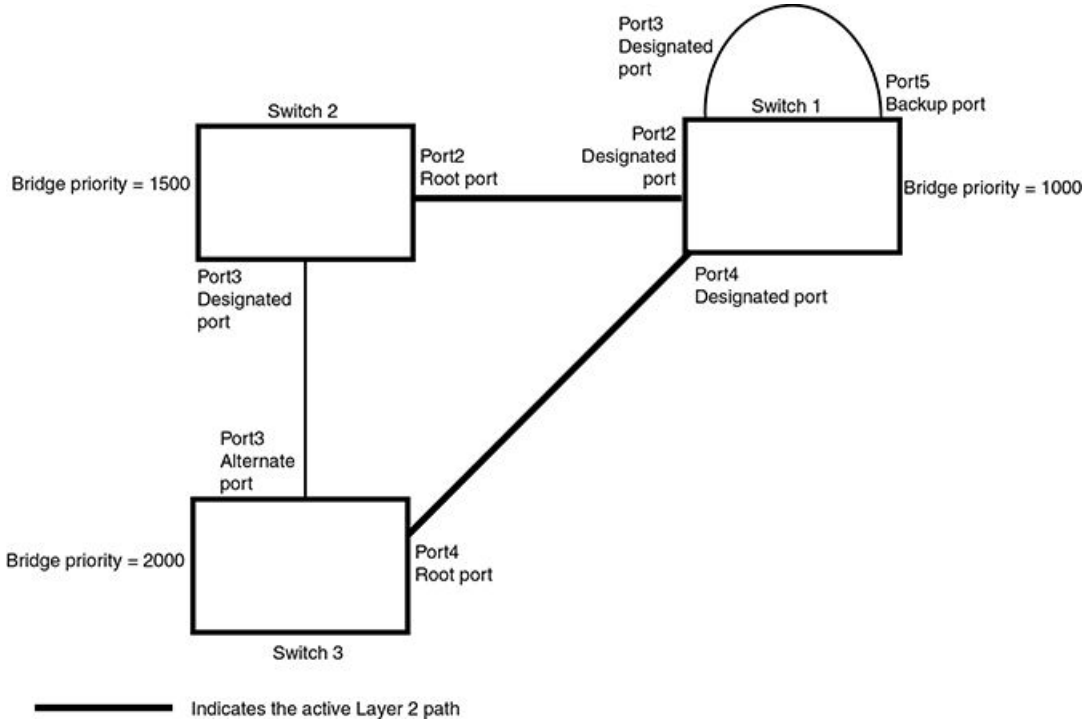
The Port2/Switch 2 bridge also sends an RST BPDU with an agreed flag Port2/Switch 1 that Port2 is the new Root port. Both ports go into forwarding states.

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 91.

FIGURE 89 Active Layer 2 path

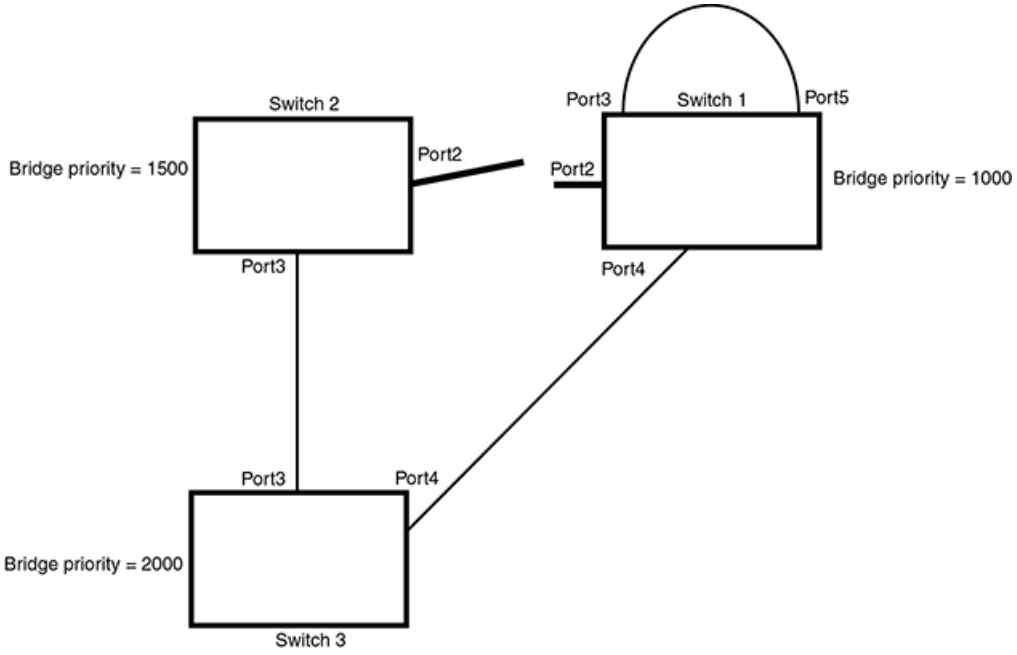


Convergence after a link failure

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 92).

FIGURE 90 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.

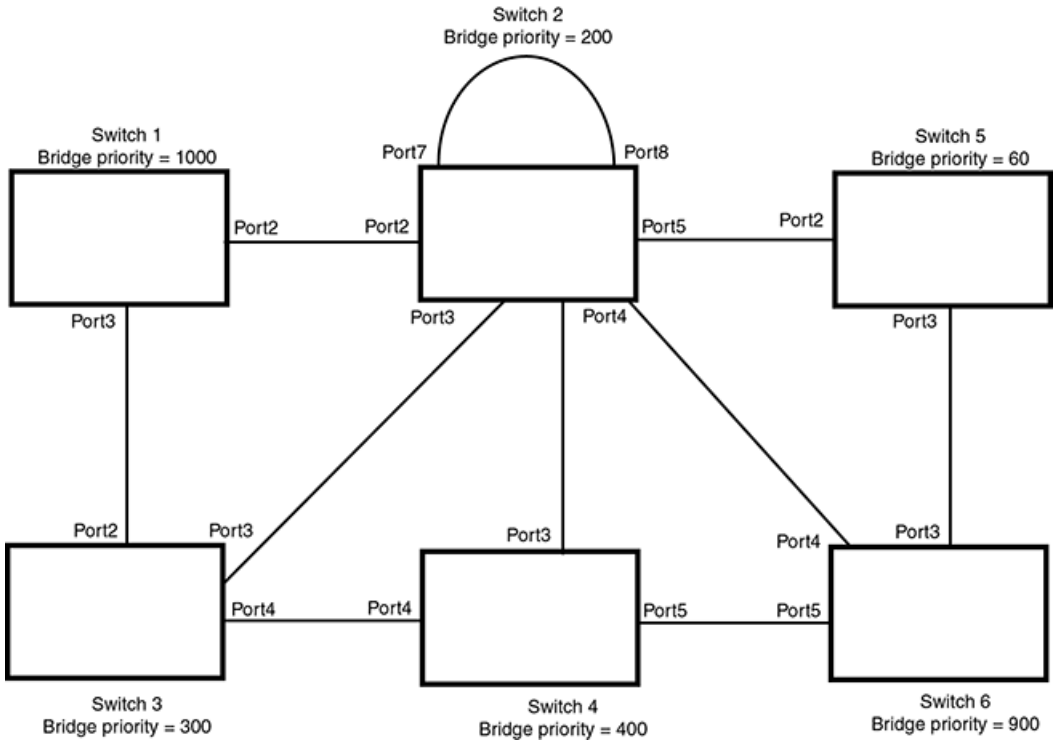
Now Port3/Switch 2 does not go into a forwarding state instantly like the Root port. It waits until the forward delay timer expires twice on that port while it is still in a Designated role, before it can proceed to the forwarding state. The wait, however, does not cause a denial of service, since the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

When fully restored, the topology is the same as that shown on Convergence at start up on page 407.

Convergence in a complex RSTP topology

The following is an example of a complex RSTP topology.

FIGURE 91 Complex RSTP topology



In Figure 93, 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.

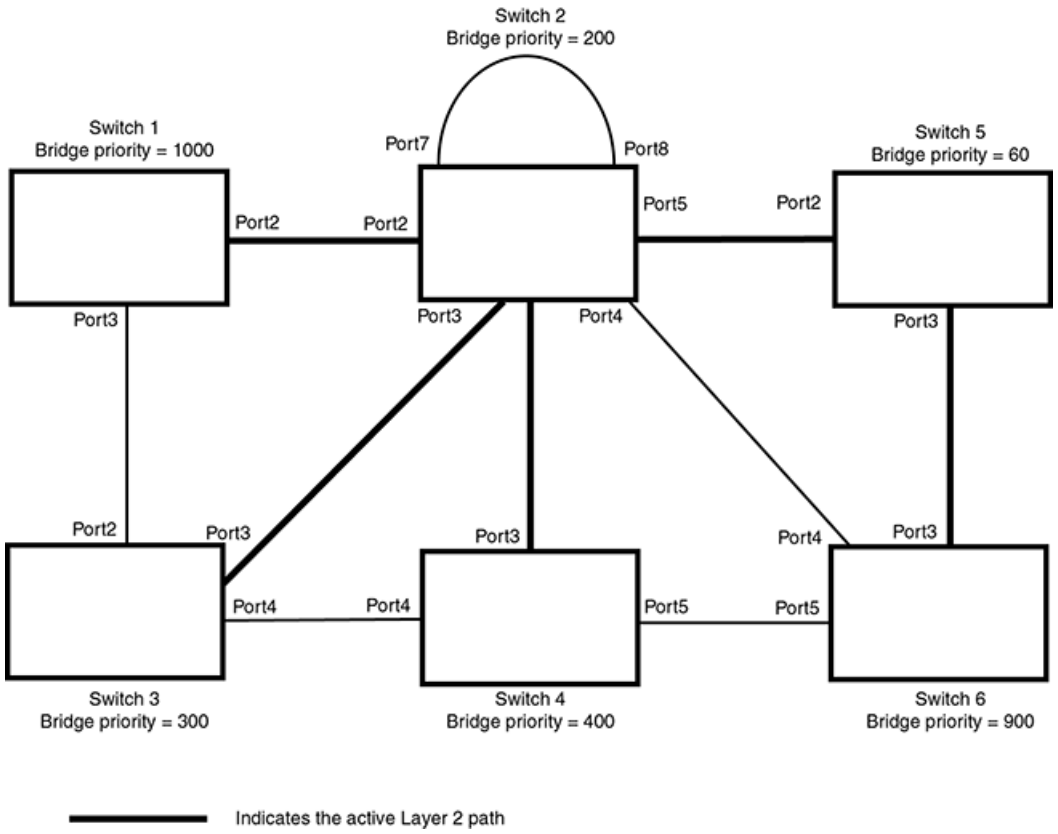
Port4/Switch 3 receives an RST BPDU that is inferior to what it can transmit; therefore, the port retains its Designated port role.

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 94 shows the active Layer 2 path of the topology in Figure 93.

FIGURE 92 Active Layer 2 path in complex topology



Propagation of topology change

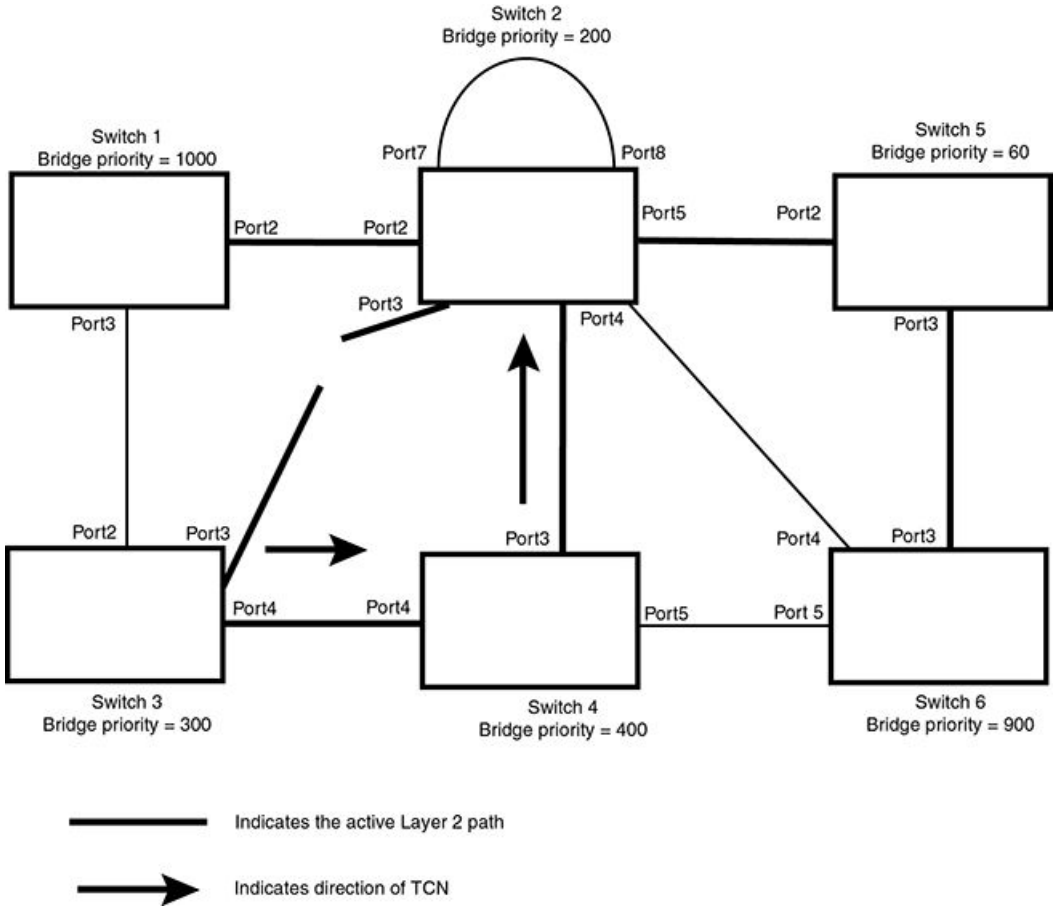
The Topology Change state machine generates and propagates the topology change notification messages on each port. When a Root port or a Designated port goes into a forwarding state, the Topology Change state machine on those ports send a topology change notice (TCN) to all the bridges in the topology 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 95, 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 95.)

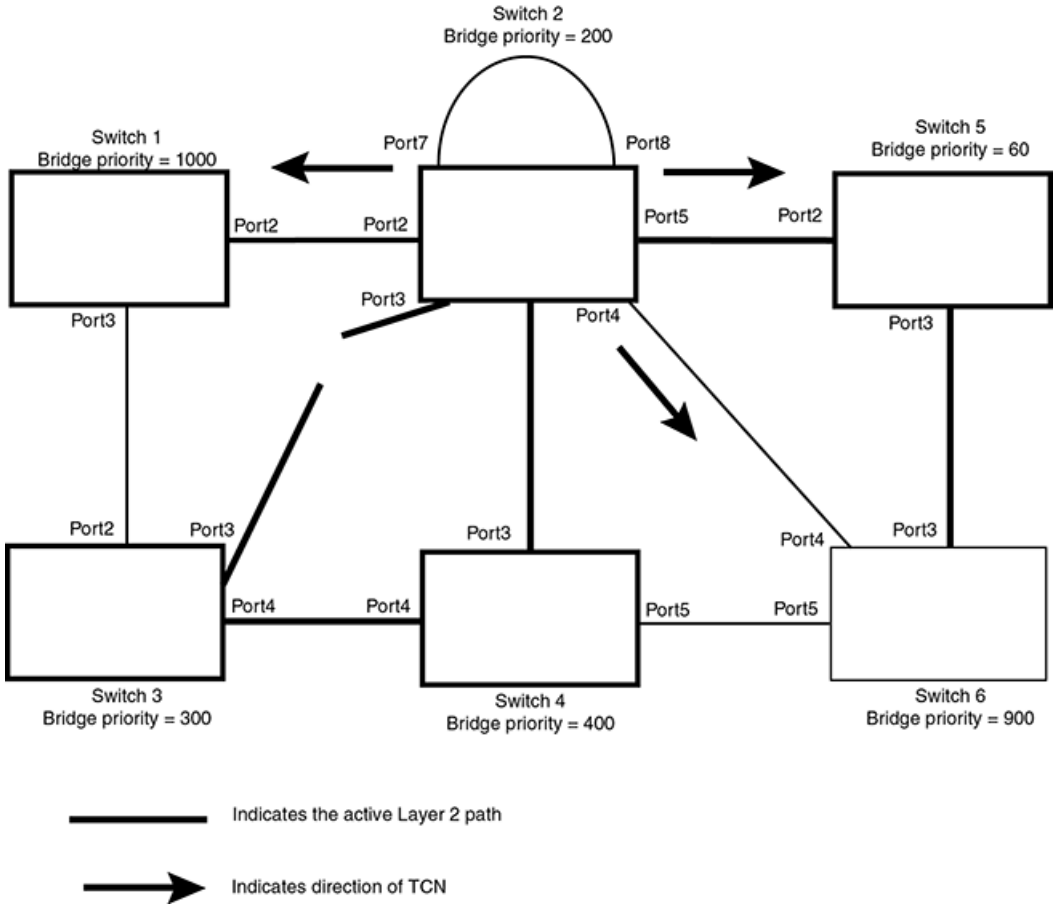
FIGURE 93 Beginning of topology change notice



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

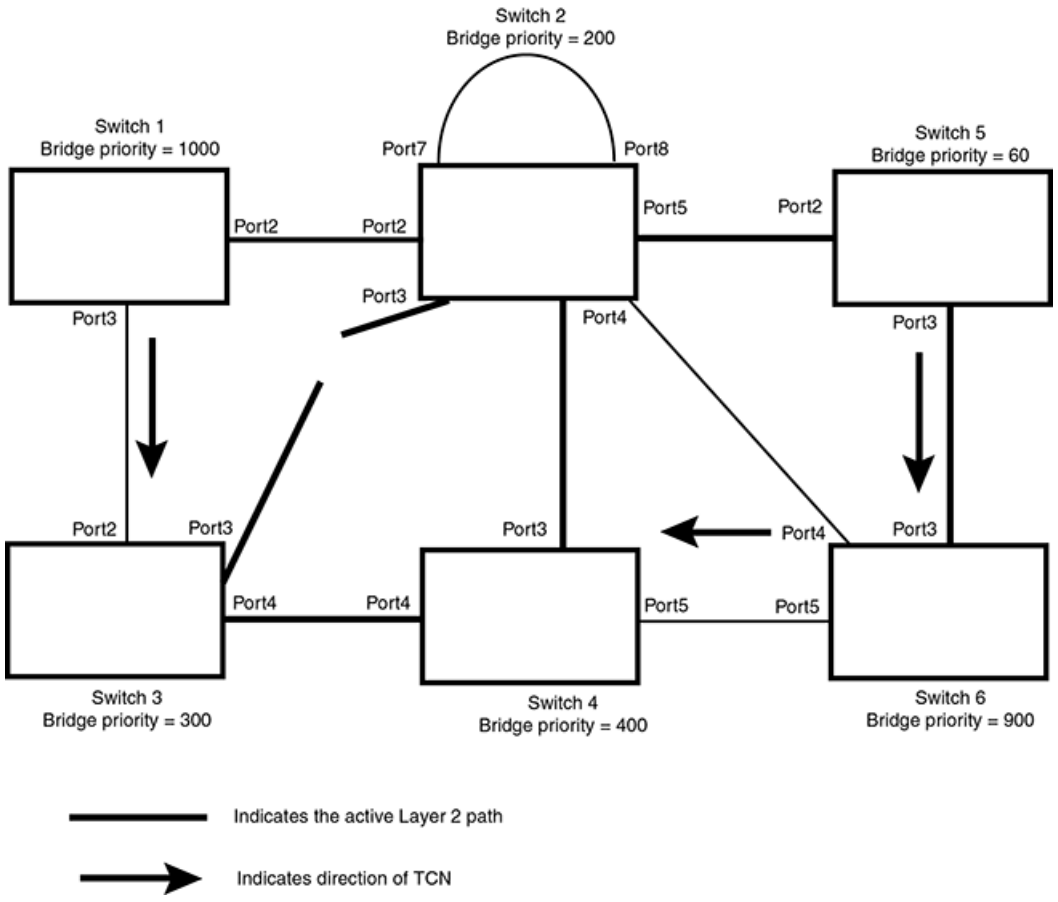
- Port5/Switch 2 sends the TCN to Port2/Switch 5
- Port4/Switch 2 sends the TCN to Port4/Switch 6
- Port2/Switch 2 sends the TCN to Port2/Switch 1

FIGURE 94 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 95 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.

NOTE
Intermixing the two types of bridges in the network topology is not advisable if you want to take advantage of the rapid convergence feature.

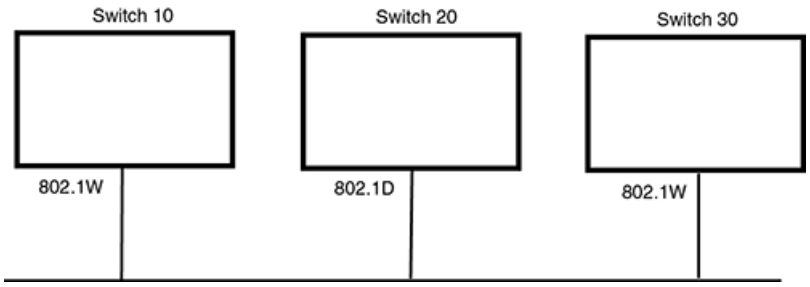
Compatibility with 802.1D means that an RSTP-enabled port can send BPDUs in the STP or 802.1D format when one of the following events occur:

- 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 98, 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 96 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 Brocade device.

You can enable or disable RSTP at the following levels:

- **Port-based VLAN** - Affects all ports within the specified port-based VLAN. When you enable or disable RSTP within a port-based VLAN, the setting overrides the global setting. Thus, you can enable RSTP for the ports within a port-based VLAN even when RSTP is globally disabled, or disable the ports within a port-based VLAN when RSTP is globally enabled.
- **Individual port** - Affects only the individual port. However, if you change the RSTP state of the primary port in a LAG group, the change affects all ports in the LAG group.

RSTP in a LAG

The RSTP standard indicates that by default the path cost is determined by link speed. For a port having 1G the path cost is 20,000 and for 10G the path cost is 2,000. However, if a LAG is made consisting of n 1G ports where n is less than 10, the path cost remains as 20,000. The standard does not indicate pathcost explicitly for LAG interfaces or if the bandwidth is in intermediate value. Therefore, during RSTP deployment you may find that though a LAG has greater bandwidth, its in blocking/discarding state as its pathCost is same as any 1G link and the portIndex of 1G port is lower, making the LAG go into a blocking/discarding state. This behavior is not restricted to 1G or 10G link speed but span across different link speeds. The same behavior also holds TRUE for STP deployments.

Enabling or disabling RSTP in a port-based VLAN

Use the following procedure to disable or enable RSTP on a Brocade device on which you have configured a port-based VLAN. Changing the RSTP state in a VLAN affects only that VLAN.

To enable RSTP for all ports in a port-based VLAN, enter commands such as the following.

```
device(config)# vlan 10
device(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.

```
device(config)# rstp single
```

Syntax: [no] rstp single

Disabling or enabling RSTP on a port

The **rstp** command must be used to initially enable RSTP on ports. Both commands enable RSTP on all ports that belong to the VLAN or to the single spanning tree.

Once RSTP is enabled on a port, it can be disabled on individual ports. RSTP that have been disabled on individual ports can then be enabled as required.

> NOTE
> If you change the RSTP state of the primary port in a LAG group, the change affects all ports in that LAG group.

To disable or enable RSTP on a port, enter commands such as the following.

```
device(config)# interface 1/1
device(config-if-e1000-1/1)# no spanning-tree
```

Syntax: [no] spanning-tree

Configuring maximum number of RSTP instances

Brocade devices support the **system-max rstp** command to configure the maximum number of supported RSTP instances on a system.

[no] system-max rstp *number of instances*

The *number of instances* variable indicates the maximum number of RSTP instances that can be configured on the device. The valid number of instances are 1 through 256. The default number is 32 instances.

> NOTE
> Before you downgrade from Brocade NetIron Release 5.9 to a lower release and restart the device, it is recommended that you reduce the number of RSTP instances to 128 or a lower value using the **system-max rstp** command. However, if you upgrade from Brocade NetIron Release 5.8 (or previous releases) to 5.9 and restart, there is no change in the RSTP configuration or operation since the lower number of RSTP instances are anyway supported.

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.

```
device(config)# vlan 20
device(config-vlan-20)# rstp priority 0
```

To make this change in the default VLAN, enter the following commands.

```
device(config)# vlan 1
device(config-vlan-1)# rstp priority 0
```

Syntax: [**rstp forward-delay** *value*] | [**hello-time** *value*] | [**max-age** *time*] | [**force-version** *value*] | [**priority** *value*]

The **forward-delay** *value* parameter specifies how long a port waits before it forwards an RST BPDU after a topology change. Possible values: 4 - 30 seconds. The default is 15 seconds.

The **hello-time** *value* parameter specifies the interval between two hello packets. Possible values: 1 - 10 seconds. The default is 2 seconds.

The **max-age** *value* parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. Possible values: 6 - 40 seconds. The default is 20 seconds.

The value of **max-age** must be greater than the value of **forward-delay** to ensure that the downstream bridges do not age out faster than the upstream bridges (those bridges that are closer to the root bridge).

The **force-version** *value* parameter forces the bridge to send BPDUs in a specific format. You can specify one of the following values:

- 0 - The STP compatibility mode. Only STP (or legacy) BPDUs will be sent.
- 2 - The default. RST BPDUs will be sent unless a legacy bridge is detected. If a legacy bridge is detected, STP BPDUs will be sent instead.

The **priority** *value* parameter specifies the priority of the bridge. You can enter a value from 0 - 65535. A lower numerical value means a the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

You can specify some or all of these parameters on the same command line.

Changing port parameters

The RSTP port commands can be enabled on individual ports or on multiple ports, such as all ports that belong to a VLAN.

The RSTP port parameters are preconfigured with default values. If the default parameters meet your network requirements, no other action is required.

You can change the following RSTP port parameters using the following methods.

```
device(config)# vlan 10
device(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 59 shows the recommended path cost values from the IEEE standards.

TABLE 59 Recommended path cost values of RSTP

Link speed	Recommended (default) RSTP path cost values	Recommended RSTP path cost range
Less than 100 kilobits per second	200,000,000	20,000,000 - 200,000,000
1 Megabit per second	20,000,000	2,000,000 - 200,000,000
10 Megabits per second	2,000,000	200,000 - 200,000,000
100 Megabits per second	200,000	20,000 - 200,000,000
1 Gigabit per second	20,000	2,000 - 200,000,000
10 Gigabits per second	2,000	200 - 20,000
100 Gigabits per second	200	20 - 2,000
1 Terabits per second	20	2 - 200
10 Terabits per second	2	1 - 20

The **priority** *value* parameter specifies the preference that RSTP gives to this port relative to other ports for forwarding traffic out of the topology. You can specify a value from 0 - 240, in increments of 16. If you enter a value that is not divisible by four, the software rounds to the nearest value that is divisible by four. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.

Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to sent one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

Syslogs for RSTP

By default, syslog messages for RSTP are enabled. To disable syslogs generated by a Topology Change Notice (TCN) for RSTP, enter the following command.

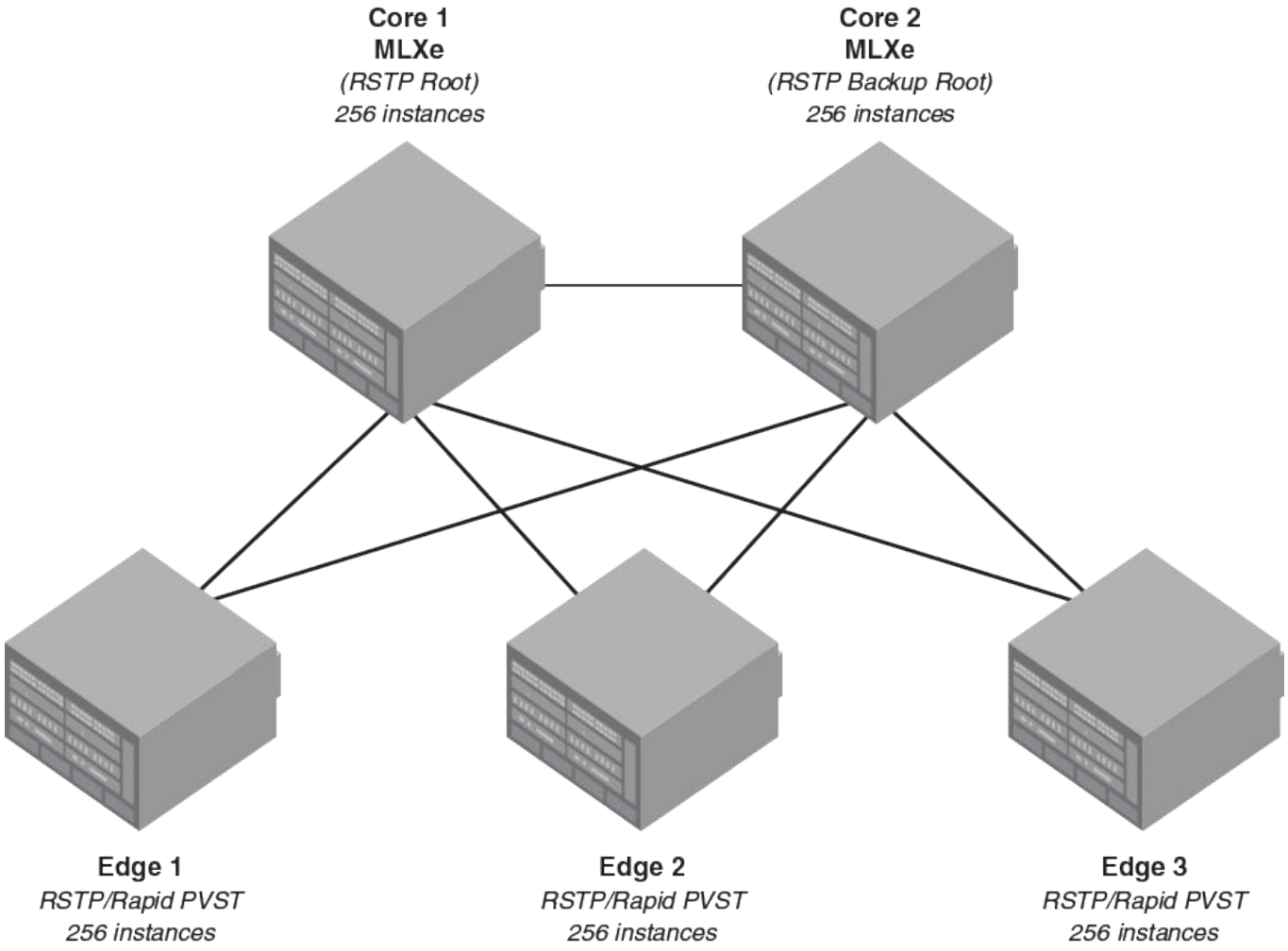
```
device(config)# no logging enable rstp tc-rx
```

Syntax: [no] logging enable rstp tc-rx

RSTP scaling recommendations and best practices

RSTP scaling recommendations and best practices are described in the following sections.

FIGURE 97 Brocade RSTP deployment example



Consider the RSTP deployment example as shown in the figure. The topology consists of two Brocade NetIron MLXe devices installed with MR2 management modules that are deployed as core devices configured with a maximum of 256 RSTP instances. Each core device is connected to multiple edge devices. An edge device can be either Brocade FastIron SX, or Brocade NetIron MLXe-MR2 device or any vendor switch that is configured with Per VLAN STP/RSTP instances or Cisco® PVST/PVST+/Rapid-PVST/Rapid-PVST + instances. The Core 1 device is configured as RSTP root node for all VLANs with the lowest RSTP priority. The Core 2 device is configured as RSTP backup root node with the next lowest RSTP topology in the entire topology.

NOTE

RSTP scaling upto 256 instances is supported on Multi-Service IronWare R05.9.00 and later software versions.

For best results with 256 RSTP instances on the NetIron device, Brocade recommends the following best practices.

- Use only Brocade BR-MLX-10GX8-X, NI-MLX-10GX8-M, NI-MLX-10GX8-D, BR-MLX-100GX2-X, BR-MLX-40GX4-M, BR-MLX-100GX1-X, BR-MLX-10GX20-X2, BR-MLX-10GX20-M, BR-MLX-100GX2-CFP2-X2, BR-MLX-100GX2-CFP2-M, BR-MLX-10GX4-IPSEC-M, and later modules for deployment of RSTP scaling. The 48x1G, 24x1G, 20x1G, and lower generation modules are not supported.
- Use RSTP with default timers. Configuring aggressive values for RSTP timers such as 'hello-time' may delay the convergence due to faster protocol timeouts.
- Configure a maximum of 256 VLANs with RSTP with maximum of 128 ports in a VLAN and a maximum of 256 VLANs for each port.

  NOTE
  RSTP may not work with 256 VLANs with 128 ports configured on each VLAN due to limit on the system resources such as memory, message queues and others. For achieving the desired effective virtual ports, topology groups are recommended to be used with RSTP.

- Configure the **wait-for-all-cards** command at the global configuration mode on the core nodes so that the UP port events are received only when all the line cards are booted up. This configuration helps in avoiding unnecessary protocol convergence and temporary loops, during node reload scenarios, due to line cards booting up in a different order.
- Configure the **rstp admin-pt2pt-mac** command for RSTP enabled non-edge ports used for interconnecting nodes which enables rapid forwarding for the ports and avoids extra MAC address flushes.
- Configure the **rstp admin-edge** command for RSTP edge ports connected to the destination nodes such as traffic generators, PCs, or VoIP devices for rapid forwarding of these ports.
- Configure BUM (Broadcast, Unknown Unicast and Multicast) rate limiting on all RSTP non-edge ports such that only 2000 to 4000 packets could reach the CPU. This avoids too many BUM packets hitting on CPU making it available for protocol convergence.
- Configure the **rl-cpu-copy** command on all RSTP non-edge ports to limit the number of packets reaching CPU for source address learning. This helps in lesser traffic loss during MAC address flush operation during RSTP convergence.
- Configure the ports of the interconnected links between core nodes in a multi-slot LAG. These ports should belong to a separate line card than the ports connected to the edge nodes. This helps the BPDUs from the root node to reach the backup root node faster for better convergence than being queued with the BPDUs from the edge nodes.
- Configure multiple links between any two nodes (core to core or core to edge) in a single LAG for load balancing and faster convergence. Any redundant links configured between two nodes cause delay in convergence.
- Configure the lowest port path cost for the link between the two core devices such that this link always remains in the forwarding state even if any of the edge node with lower RSTP bridge priority tries to act as the root bridge.
- For better RSTP convergence, configuring loop detection is not recommended when there are a large number of VLANS or VLAN groups.
- Configuring the spanning-tree root-protect command on the RSTP non-edge ports is not recommended as this may delay convergence in a few deployments.
- Ports of different bandwidths should not be configured and deployed in a LAG.
- Configure bidirectional forwarding detection (BFD) with Tx and Rx interval of 250 ms timeout with multiplier 3 with maximum of 40 sessions per line card and maximum of 250 sessions across the system. This configuration helps to avoid BFD flaps when RSTP is converged and stable.
- Configure LACP with default (long) timers. Short timers may increase the convergence time due to LACP flaps.
- Configuring STP or MSTP instances with more than 128 instances of RSTP is not supported.
- Configuring more than 128 instances of RSTP over MCT VLANs is not supported.

- Configuring MRP or ERP or VSRP, or OAM protocols such as CFM or UDLD along with more than 128 instances of the RSTP is not supported.

Displaying RSTP information

You can display a summary or details of the RSTP information.

To display a summary of RSTP, use the following command.

```
device(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
```

NOTE

After deploying or undeploying an MCT cluster, the syslog message for the final state change of the RSTP instance is not correctly updated and displayed on the console.

To display a summary of ports blocked by RSTP, use the following command.

```
Brocade#
show rstp blocked vlan 20

VLAN 20 - 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
80000024389e2d20 20     2      15     Default  3

RootBridge       RootPath  DesignatedBridge Root  Max Hel Fwd
Identifier       Cost      Identifier       Port  Age lo  Dly
hex                        hex                    sec sec sec
80000024388f6b20 2000      80000024388f6b20 3/5   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
3/6   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b20
3/7   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b20
3/8   128 2000      F   F    ALTERNATE  DISCARDING 0         80000024388f6b20
```

Syntax: show rstp [**blocked**] [**vlan vlan-id**]

The **blocked** parameter displays blocked ports only, for VLANs enabled with RSTP. When the blocked parameter is not specified, all RSTP port states are displayed.

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 60.

TABLE 60 CLI display of RSTP summary

This field...	Displays...
VLAN ID	The port-based VLAN that owns the STP instance and the number of RSTP instances on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on this device, all RSTP information is for VLAN 1.
Bridge IEEE RSTP Parameters	
Bridge Identifier	The ID of the bridge.
Bridge Max Age	The configured max age for this bridge. The default is 20.
Bridge Hello	The configured hello time for this bridge.The default is 2.
Bridge FwdDly	The configured forward delay time for this bridge. The default is 15.
Force-Version	The configured force version value. One of the following value is displayed: • 0 - The bridge has been forced to operate in an STP compatibility mode. • 2 - The bridge has been forced to operate in an RSTP mode. (This is the default.)
txHoldCnt	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Root Bridge Parameters:	
Root Bridge Identifier	ID of the Root bridge that is associated with this bridge
Root Path Cost	The cost to reach the root bridge from this bridge. If the bridge is the root bridge, then this parameter shows a value of zero.
Designated Bridge Identifier	The bridge from where the root information was received.It can be from the root bridge itself, but it could also be from another bridge.
Root Port	The port on which the root information was received. This is the port that is connected to the Designated Bridge.
Max Age	The max age is derived from the Root port. An RSTP-enabled bridge uses this value, along with the hello and message age parameters to compute the effective age of an RST BPDU. The message age parameter is generated by the Designated port and transmitted in the RST BPDU. RST BPDUs transmitted by a Designated port of the root bridge contains a message value of zero. Effective age is the amount of time the Root port, Alternate port, or Backup port retains the information it received from its peer Designated port. Effective age is reset every time a port receives an RST BPDU from its Designated port. If a Root port does not receive an RST BPDU from its peer Designated port for a duration more than the effective age, the Root port ages out the existing information and recomputes the topology. If the port is operating in 802.1D compatible mode, then max age functionality is the same as in 802.1D (STP).
Hello	The hello value derived from the Root port. It is the number of seconds between two Hello packets.
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

TABLE 60 CLI display of RSTP summary (continued)

This field...	Displays...
	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 391 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 395 and Edge port and non-Edge port states on page 395.
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.

```
device(config)#show rstp detail
VLAN 10 - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:
BridgeId 0001000480a04000, RootBridgeId 0001000480a04000
Control ports - ethernet 1/3 ethernet 1/13
```

```
ForceVersion 2, MigrateTime 3, TxHoldCount 3
RSTP (IEEE 802.1w) Port Parameters:
Port 1/3 - Role: DISABLED - State: DISABLED
Port 1/13 - Role: DISABLED - State: DISABLED
```

Syntax: show rstp detail [**vlan vlan-id**]

The **vlan** *vlan-id* parameter displays RSTP information for the specified port-based VLAN.

The **show RSTP detail** command shows the following information.

This field...	Displays...
VLAN ID	ID of the VLAN that owns the instance of RSTP and the number of RSTP instances on that VLAN.
Bridge ID	ID of the bridge.
Control ports	Ports assigned to the VLAN
forceVersion	the configured version of the bridge: • 0 - The bridge has been forced to operate in an STP compatible mode. • 2 - The bridge has been forced to operate in an RSTP mode.
MigrateTime	The number of seconds the bridge took to migrate from STP to RSTP mode.
txHoldCount	The number of BPDUs that can be transmitted per Hello Interval. The default is 3.
Port	ID of the port in slot#/port# format.
Role	The current role of the port: • Root • Designated • Alternate • Backup • Disabled Refer to Bridges and bridge port roles on page 391 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 395 and Edge port and non-Edge port states on page 395.

Configuring RSTP under an ESI VLAN

RSTP can also be configured under a VLAN that is part of a user-configured ESI. For example, to enable RSTP on a VLAN that is part of an ESI, configure the following commands.

```
device(config)# esi customer1 encapsulation cvlan
device(config-esi-customer1)# vlan 100
device(config-esi-customer1-vlan-100)# rstp
```

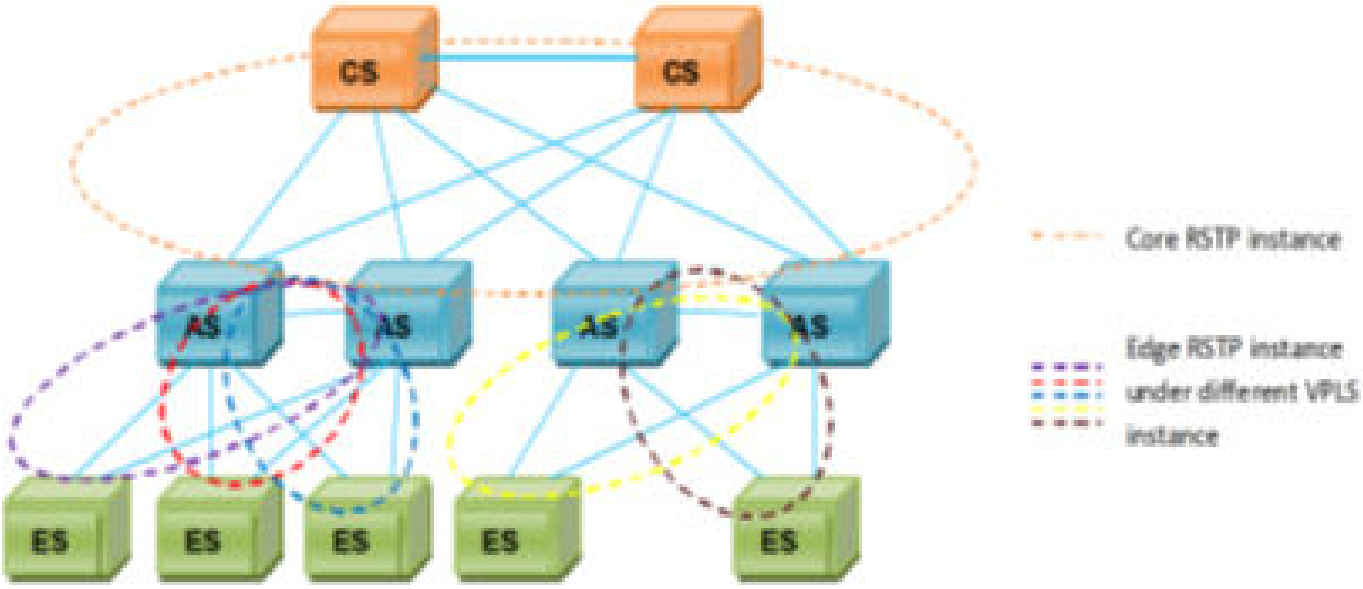
RSTP support for PB and PBB

A PBB network is comprised of a set of Backbone Core Bridges (BCBs) and Backbone Edge Bridges (BEBs). BEBs are interconnected by some or all of the S-VLANs supported by a PB network. Each BEB provides interfaces that encapsulate customer frames, thus allowing customer MAC addresses (C-MAC) and VLANs (S-VLAN) to be independent of backbone MAC addresses (B-MAC) and VLANs (B-VLAN) used to relay those frames across the backbone.

In Brocade NetIron CES Series and Brocade NetIron CER Series, RSTP over PBB is supported in previous releases implemented using the ESI framework.

Consider a scenario where all BEBs (AS - Access Switch) are connected to two BCBs (CS - Core Switch) to provide dual home for all the BEBs, and all PBs (ES - Edge Switch) are connected to two BEBs to provide dual home for all the PBs in the network. With the dual homing of the ASs to the CSs and ESs to the ASs, all failures are protected. This dual homing support creates potential loops in a PB network and PBB network. To achieve the functionality of dual homing of AS and ES, the active topology of a PB network and PBB network area should be isolated by running different instance of RSTP.

FIGURE 98 RSTP isolation in dual homing of AS and ES



The network shown in Figure 100 has two Core Switches (CSs) to provide resiliency in the core. The CS functions as a Backbone Core Bridge (BCB). All Access Switches (AS) in the network dual home to the two CSs. The AS functions as a Backbone Edge Bridge (BEB). The CSs do not have any service interfaces. The function of the CS is to switch traffic between the ASs. As a BCB, a CS will switch on the outer PBB B-tag and will not perform any PBB encapsulation.

The AS and CS switches in the network will form a single RSTP region. With the dual homing of the ASs to the CSs, all failures are protected against. ES dual homing will help to protect the failure of AS.

Core RSTP

RSTP will run in the PBB core for loop detection and avoidance. All ASs and CSs will participate in the RSTP and one CS will be selected as the root for the core RSTP instance. The assumption is there will be only one RSTP instance running in the PBB core and traffic flow will be through one CS which is the root bridge.

A backbone service provider can either use RSTP or MSTP. Preferably MSTP shall be used, since this will allow the service provider to use different active paths for different B-VLANs.

To avoid a potential loop in the core PBB network, RSTP will be enabled. The regular VLAN corresponds to VPLS B-VLAN in AS/BEB bridges.

Edge RSTP

Dual homing ES to two ASs would require RSTP to avoid loops. An AS will have several RSTP instances provisioned on the ES facing side. Here one AS will be acting as the root and the Edge RSTP is completely separate and independent from the Core RSTP.

The Edge RSTP instance should be enabled on the VPLS instance on AS/ES.

For dual homing of the ES, the ASs can be connected using either of the following two methods.

1. Two AS/BEB switches connected via the S-tagged endpoint.
2. Two AS/BEB switches connected via the IB-tagged endpoint.

   In each case, the RSTP behavior is different, which is explained below.

   Figure 101 shows that BEB-1, BEB-2 and PB could be of same S-VLAN or different S-VLANs. Since RSTP is enabled under the VPLS instance, all the VPLS VLANs which belong to that VPLS instance will be considered as part of same RSTP instance. BPDUs will be transmitted on all the VPLS endpoints with the VLAN tag associated with that end point.

   FIGURE 99 RSTP convergence when two AS connected via S tagged endpoint

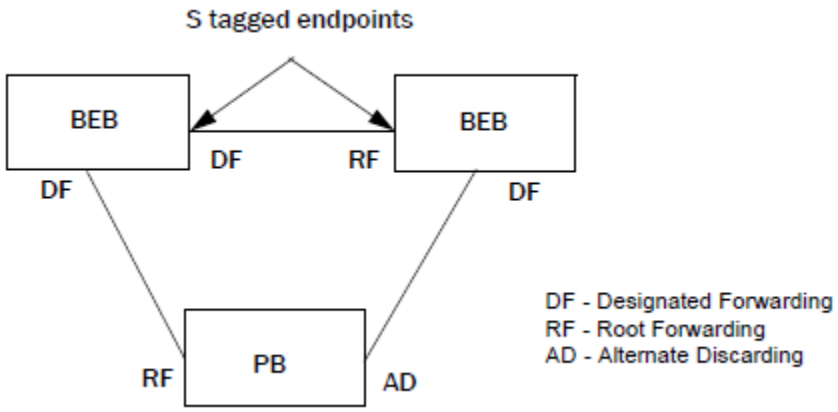
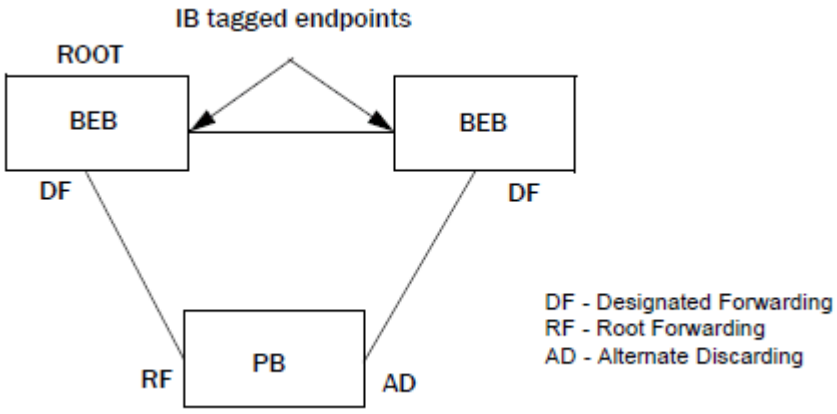


   Figure 102 shows that BEB-1 and BEB-2 are connected via an IB tagged endpoint. The topology convergence is the same as in Figure 101. The difference comes in the BPDU transmitted out of the IB tagged endpoint that will be a tunneled packet which will have a PBB header. In this topology the IB- tagged end point should be always in the forwarding state by configuring either BEB-1 or BEB-2 as a ROOT bridge.

FIGURE 100 RSTP convergence when two AS connected via IB tagged endpoint



BPDU behavior on VPLS endpoints

1. By default, the BPDU generated from the VPLS end point will be with destination MAC of 00-00-00-00-00-08.
2. Upon receiving a BPDU with destination MAC 00-00-00-00-00-00, the RSTP will start responding with that MAC on next transmitted BPDUs. (This is applicable for the VPLS VLAN acting as a C-VLAN or S-VLAN.)
3. The VPLS B-VLAN end point will not respond for a tunneled BPDU with the STP MAC 00-00-00-00-00-00. It will be considered as a tunneled BPDU from customer bridges and be tunneled across S-VLAN as well as C-VLAN.
4. The VPLS B-VLAN end point will be consuming the BPDU. Only when RSTP is enabled on the IB tagged end point and received tunneled BPDU, it has a PBB-STP MAC. If RSTP is not enabled, the received BPDU with PBB-STP MAC will be send to S-VLAN

Limitations

- Total RSTP instances is limited to 128, including regular VLAN and VPLS instances.
- RSTP Single is not supported in VPLS VLAN.
- The is no MIB support for PBB RSTP.
- Do not use the same VLAN ID for a regular VLAN and VPLS instance in the network. Enabling RSTP on a regular VLAN and VPLS instance which has a VPLS VLAN with the same VLAN ID as the regular VLAN ID can lead to an undesirable topology convergence.
- It is not possible to enable RSTP on a PBB VPLS instance if that instance has multiple VLANS configured with same member ports.
- RSTP interoperability between Brocade NetIron MLX Series and Brocade NetIron CER Series devices is not supported if both are acting as a BEB.
- Running RSTP on a VPLS Instance will not avoid the pure Layer 2 forwarding loop created by the regular B-VLAN in the BEBs. Run RSTP on regular B-VLAN in BEBs.
- Switchover and hitless upgrade are not supported on PBB RSTP.
- When changing the RSTP parameters on a regular B-VLAN, the parameters also need to be changed on the VPLS instance. The expectation is to have the same port states for B-VLAN (regular) end points and VPLS ISID end points.

Configuration commands

Use the following commands to configure RSTP on a PBB VPLS instance.

Syntax: [no] rstp [**forward-delay** *value*] | [**hello-time** *value*] | [**max-age** *time*] | [**force-version** *value*] | [**priority** *value*]

The **forward-delay** parameter specifies how long a port waits before it forwards an RST.

The **hello-time** parameter specifies the interval between two hello packets. The values range from 1 to 10 seconds. The default is 2 seconds.

The **max-age** parameter specifies the amount of time the device waits to receive a hello packet before it initiates a topology change. Acceptable values range from 6 to 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.

The **force-version** parameter forces the bridge to send BPDUs in a specific format. You can specify either 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** parameter specifies the priority of the bridge. You can enter a value from 0 to 65535. A lower numerical value means a the bridge has a higher priority. Thus, the highest priority is 0. The default is 32768.

The **[no]** version disables the feature and returns settings to default.

Syntax: [no] rstp etherne t *slot/portnum* **path-cost** *value* | **priority** *value* | [**admin-edge-port**] | [**admin-pt2pt-mac**] | [**force-migration-check**]

The **ethernet***slot/portnum* parameter specifies the interface used.

The **path-cost** 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 to 20,000,000.

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 to 240, in increments of 16. If you enter a value that is not divisible by four, the software rounds to the nearest value that is divisible by four. The default is 128. A higher numerical value means a lower priority; thus, the highest priority is 8.

Set the **admin-edge-port** to enabled or disabled. If set to enabled, then the port becomes an edge port in the domain.

Set the **admin-pt2pt-mac** to enabled or disabled. If set to enabled, then a port is connected to another port through a point-to-point link. The point-to-point link increases the speed of convergence. This parameter, however, does not auto-detect whether or not the link is a physical point-to-point link.

The **force-migration-check** parameter forces the specified port to sent one RST BPDU. If only STP BPDUs are received in response to the sent RST BPDU, then the port will go return to sending STP BPDUs.

Show commands

The **show rstp** command output displays VPLS instance ID if RSTP is running in VPLS VLAN.

device(config)# show rstp

Syntax: show rstp [**vlan** *vlan-id*] [**vpls id** *vpls-id*]

```
device(config)# show rstp
VPLS Instance ID 1  - 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                   tedcost  bridge
1/3  128  20000    T   F    DISABLED DISABLED 0        0000000000000000
1/13 128  20000    T   F    DISABLED DISABLED 0        0000000000000000
```

The **show rstp detail** command displays VPLS instance ID if RSTP is running in VPLS VLAN.

```
device(config)# show rstp detail vpls id 1
```

Syntax: show rstp detail [**vlan** *vlan-id*] [**vpls id** *vpls-id*]

```
NetIron(config)#show rstp detail
VPLS Instance ID - 1  - RSTP instance 0
------------------------------------------------------------------------
RSTP (IEEE 802.1w) Bridge Parameters:
BridgeId 0001000480a04000, RootBridgeId 0001000480a04000
Control ports - ethernet 1/3 ethernet 1/13
ForceVersion 2, MigrateTime 3, TxHoldCount 3
RSTP (IEEE 802.1w) Port Parameters:
Port 1/3 - Role: DISABLED - State: DISABLED
Port 1/13 - Role: DISABLED - State: DISABLED
```

The **show mpls vpls detail** command displays the VPLS instance ID if RSTP is running in VPLS VLAN.

Syntax: show mpls vpls detail

```
NetIron #show mpls vpls id 1
VPLS as, Id 1, Max mac entries: 2048
 PBB
   Bridge Destination MAC Address: None (NHT Index: 0)
 Total vlans: 2, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
 IFL-ID: n/a
 Enabled L2 Protocol:RSTP
 Vlan 100: Topo ID 1
   Tagged: ethe 1/1

   Port    Protocol  State
   1/1     RSTP      DISABLED
   1/2     RSTP      FORWARDING
  Vlan 200
   Tagged: ethe 1/1
      CPU-Protection: OFF
   Local Switching: Enabled
   Extended Counter: ON
```

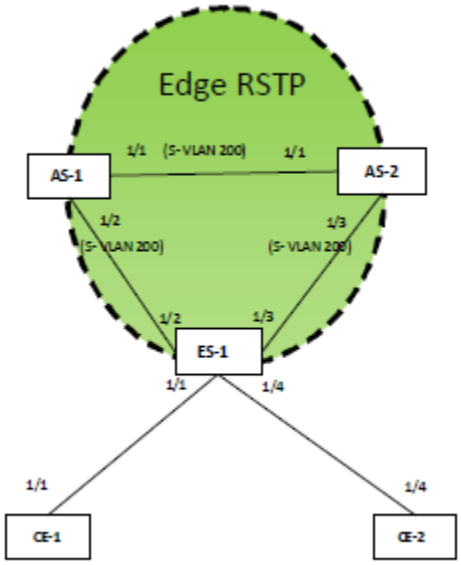
Use case scenarios

Use case 1: Edge RSTP - AS-1 is connected to AS-2 and ES-1 via S-tagged endpoints of same S-VLAN

The following deployment scenario is a case where RSTP is deployed for a single S-VLAN in a PB network. VPLS VLAN 200 (S-VLAN) is responsible for carrying traffic to the PB network. In this case, AS-1 is configured as ROOT Bridge, VPLS VLAN 200 is configured on AS-1, AS-2, and ES-1 which connects the three bridges together. RSTP is running on the VPLS Instance on AS-1, AS-2 and ES-1.

The following describes the steps to configure the nodes in the topology.

FIGURE 101 Edge RSTP topology 1



Configuring AS-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/2
```

RSTP Configuration on vpls instance 1

```
device_AS-1(config-mpls-vpls-pb-svlan)#rstp
device_AS-1(config-mpls-vpls-pb-svlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/3
```

RSTP Configuration on vpls instance 1

```
device_AS-2 (config-mpls-vpls-pb-svlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 200
device_ES-1(config-mpls-vpls-pb-vlan-vlan-200)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 300 on the customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag eth 1/1 eth 1/4
```

RSTP Configuration on VPLS instance

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 300 and add port 1/1 to it.

```
device_CE-1(config)#vlan 300
device_CE-1(config-vlan-300)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 300 and add port 1/4 to it.

```
device_CE-2(config)#vlan 300
device_CE-2(config-vlan-300)#tagged ethernet 1/4
```

If RSTP needs to be enabled in CE bridges, the following configuration should be applied.

Configuring CE-1 and CE-2

RSTP Configuration

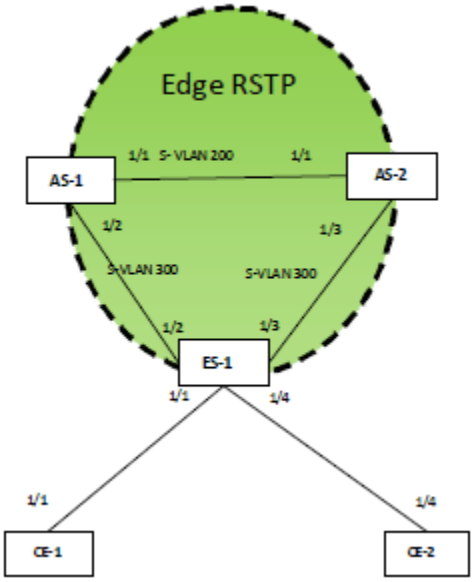
```
device_CE-1(config)#vlan 300
device_CE-1(config-vlan-300)#rstp
```

Use case 2: Edge RSTP - AS-1 is connected to AS-2 and ES-1 via S-tagged endpoints of different S-VLAN

The following deployment scenario is a case where RSTP is deployed on two different S-VLANs of the same VPLS instance in a PB network. Here AS-1 is configured as a ROOT Bridge and RSTP is running on the VPLS Instance on AS-1, AS-2, and ES-1. VPLS VLAN 200 configured on AS-1 and AS-2 acts as an S-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 configured on AS-1 and AS-2 acts as another S-VLAN which connects AS-1 and AS-2 to the ES-1. AS-1 and AS-2 has the VPLS VLAN 200 and 300 configured under same VPLS instance. VPLS VLAN 300 (S-VLAN) is configured ES-1 connects to AS-1 and AS-2.

The following discussion describes procedure required to configure the nodes in the topology.

FIGURE 102 Edge RSTP topology 2



Configuring AS-1

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
```

```
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-svlan)#rstp
device_AS-1(config-mpls-vpls-pb-svlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-2(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-svlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure the C-VLAN 400 on customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 400
device_ES-1(config-mpls-vpls-pb-vlan-vlan-400)#tag eth 1/1 eth 1/4
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

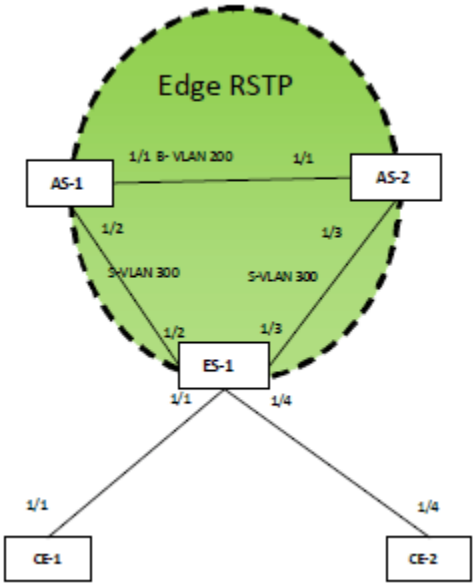
Use case 3: Edge RSTP - AS-1 is connected to AS-2 via IB-tagged endpoint and both the AS on ES facing side with same S-VLAN

The following deployment scenario is a case where RSTP is deployed on a VPLS instance which has a S-VLAN configured to the ES facing side and B-VLAN configured which connects 2 ASs.

AS-1 is configured as a ROOT Bridge and RSTP is running on the VPLS Instance on AS-1, AS-2 and ES-1. VPLS VLAN 200 is configured in AS-1 and AS-2 acts as B-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 is configured in AS-1 and AS-2 acts as S-VLAN which connects AS-1 and AS-2 to the ES-1. In AS-1 and AS-2 VPLS VLAN 200 and 300 are configured under the same VPLS instance. VPLS VLAN 300 (S-VLAN) is configured on ES-1, which connects to AS-1 and AS-2.

The following discussion describes how to configure the nodes in the topology.

FIGURE 103 Edge RSTP topology 3



Configuring AS-1

Tag type configuration

Configure port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
```

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/2
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 200
device_AS-1(config-vlan-200)#tagged ethernet 1/1
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-vlan 1
device_AS-1(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-1(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan)#rstp
device_AS-1(config-mpls-vpls-pb-vlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure the port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
:
```

Configure the port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/3
```

B-VLAN Configuration

```
device_AS-2(config)#vlan 200
device_AS-2(config-vlan-200)#tagged ethernet 1/1
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pbb-vlan 1
device_AS-2(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-2(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_AS-2(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/3
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure the port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2 ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 400 on a customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 400
device_ES-1(config-mpls-vpls-pb-vlan-vlan-400)#tag eth 1/1 eth 1/4
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

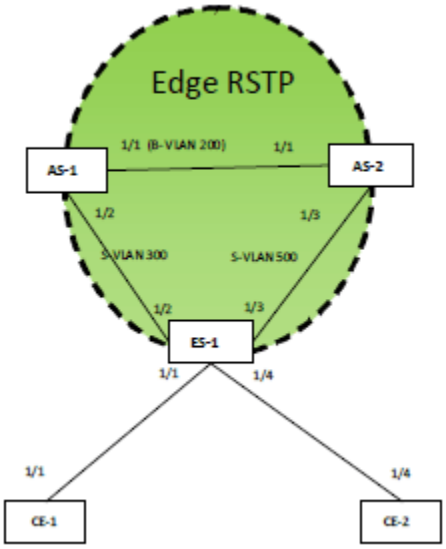
Use case 4: Edge RSTP - AS-1 is connected to AS-2 via IB-tagged endpoint and both the AS on ES facing side with different S-VLAN

The following deployment scenario is a case where RSTP is deployed on a VPLS instance which has a S-VLAN configured to the ES facing side and B-VLAN configured which connects 2 ASs.

AS-1 is configured as a ROOT Bridge. VPLS VLAN 200 is configured in AS-1 and AS-2 acts as B-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 is configured in AS-1 connects AS-1 to ES-1 and VPLS VLSN 500 on AS-2 connects AS-2 to ES-1. In AS-1 VPLS VLAN 200 and 300 are configured under same VPLS instance and AS-2 VPLS VLAN 200 and 500 are configured under the same VPLS Instance. VPLS VLAN 300 (S-VLAN) configured on ES-1 which connects to AS-1 and VPLS VLAN 500 connects to AS-2.

The following discussion describes how to configure the nodes in the topology.

FIGURE 104 Edge RSTP topology 4



Configuring AS-1

Tag type configuration

Configure the port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
```

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/2
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 200
device_AS-1(config-vlan-200)#tagged ethernet 1/1
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-vlan 1
device_AS-1(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-1(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-vlan)#rstp
device_AS-1(config-mpls-vpls-pb-vlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure a port tag type for B-VLAN to 0x88a8.

```
device_AS-1(config)#tag-type 88e8 eth 1/1
```

Configure a port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/3
```

B-VLAN Configuration

```
device_AS-2(config)#vlan 200
device_AS-2(config-vlan-200)#tagged ethernet 1/1
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pbb-vlan 1
device_AS-2(config-mpls-vpls-pbb-vlan)#pbb
```

B-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pbb-vlan-pbb)#vlan 200 isid 101010
device_AS-2(config-mpls-vpls-pbb-vlan-vlan-200-isid-101010)#tagged ethernet 1/1
```

S-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan-pbb)#vlan 500
device_AS-2(config-mpls-vpls-pb-vlan-vlan-500)#tag ethernet 1/3
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-vlan)#rstp
```

Configuring ES-1

Tag type configuration

Configure a port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 300.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-vlan 1
device_ES-1(config-mpls-vpls-pb-vlan)#pbb
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 300
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/2
device_ES-1(config-mpls-vpls-pb-vlan-pbb)#vlan 500
device_ES-1(config-mpls-vpls-pb-vlan-vlan-300)#tag ethernet 1/3
```

C-VLAN Configuration

Configure C-VLAN 400 on customer port.

```
device_ES-1(config-mpls-vpls-pb-vlan)#vlan 400
device_ES-1(config-mpls-vpls-pb-vlan-vlan-400)#tag eth 1/1 eth 1/4
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-vlan)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

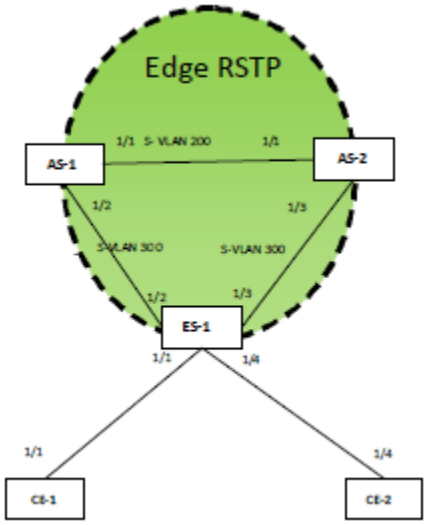
Use case 5: Edge RSTP- Interoperability with Brocade NetIron CES Series and Brocade NetIron CER Series

In this scenario AS-1 is connected to AS-2 and ES-1 via S-tagged endpoint of different S-VLAN, and ES-1 is a Brocade NetIron CES Series and Brocade NetIron CER Series device.

The following scenario is a case where RSTP is deployed on two different S-VLANs of the same VPLS instance in a PB network. AS-1 is configured as a ROOT Bridge and RSTP is running on the VPLS Instance on AS-1, AS-2, and ES-1. VPLS VLAN 200 is configured on AS-1 and AS-2 acts as a S-VLAN which connects AS-1 and AS-2. VPLS VLAN 300 is configured on AS-1 and AS-2 acts as another S-VLAN which connects AS-1 and AS-2 to the ES-1. AS-1 and AS-2 has VPLS VLAN 200 and 300 configured under same VPLS instance. VPLS VLAN 300 (S-VLAN) is configured with ES-1 and connects to AS-1 and AS-2.

The following discussion describes how to configure the nodes in the topology.

FIGURE 105 Edge RSTP topology 5



Configuring AS-1

Tag type configuration

Configure a port tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-1(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-svlan)#rstp
device_AS-1(config-mpls-vpls-pb-svlan)#rstp priority 100
```

Configuring AS-2

Tag type configuration

Configure a port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 300
device_AS-2(config-mpls-vpls-pb-svlan-vlan-300)#tag ethernet 1/2
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pb-svlan)#rstp
```

Configuring ES-1

Port-type configuration

```
device_(config)# interface ethernet 1/2
device_(config-if-e1000-1/2)# port-type provider-network
device_(config-if-e1000-1/2)# enable
device_(config)# interface ethernet 1/3
device_(config-if-e1000-1/3)# port-type provider-network
device_(config-if-e1000-1/3)# enable
```

Port-type configuration

```
device_(config)# interface ethernet 1/1
device_(config-if-e1000-1/1)# port-type customer-edge
device_(config-if-e1000-1/1)# enable
device_(config)# interface ethernet 1/4
device_(config-if-e1000-1/4)# port-type customer-edge
device_(config-if-e1000-1/4)# enable
device_(config)# esi provider encapsulation svlan
device_ (config-esi-provider)# vlan 300
```

```
device_(config-esi-provider-vlan-300)# tagged ethernet 1/2
device_ (config-esi-provider-vlan-300)# tagged ethernet 1/3
device_ (config-esi-provider-vlan-300)# exit
device_ (config-esi-provider)# exit
device_(config)# esi customer encapsulation cvlan
device_(config-esi- customer )# vlan 400
device_(config-esi- customer -vlan-400)# tagged ethernet 1/1
device_(config-esi- customer -vlan-400)# tagged ethernet 1/4
device_(config-esi- customer -vlan-400)# exit
device_(config-esi- customer)# exit
```

RSTP Configuration

```
device_(config)# esi customer encapsulation  cvlan
device_(config-esi- customer )# vlan 400
device_(config-esi- customer )#rstp
device_(config)# esi provider encapsulation  svlan
device_(config-esi-provider)# vlan 300
device_(config-esi-provider-vlan-300)#rstp
```

Configuring CE-1

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/1 to it.

```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#tagged ethernet 1/1
```

Configuring CE-2

C-VLAN Configuration

Configure a regular Layer 2 VLAN with 400 (C-VLAN) and add port 1/4 to it.

```
device_CE-2(config)#vlan 400
device_CE-2(config-vlan-400)#tagged ethernet 1/4
```

Configuring RSTP on CE-1 and CE-2

RSTP Configuration

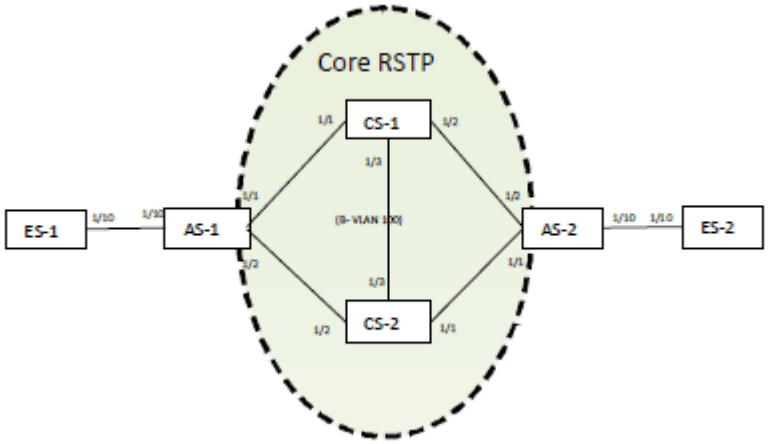
```
device_CE-1(config)#vlan 400
device_CE-1(config-vlan-400)#rstp
```

Use case 6: Core RSTP

The following deployment scenario is a case where RSTP is deployed for a single B-VLAN in a PBB network. In CS-1 and CS-2 a regular VLAN 100 is configured as B-VLAN which carries the traffic to the PBB core network. AS-1 and AS-2 have a regular VLAN and VPLS VLAN 100 which carries PBB traffic towards the B-VLAN. CS-1 is configured as the ROOT Bridge and RSTP is running over regular VLAN 100 in CS-1 and CS-2. In AS-1 and AS-2 RSTP runs over the regular VLAN 100 corresponds to the VPLS VLAN B-VLAN 100.

The following discussion describes how to configure the nodes in the topology.

FIGURE 106 Core RSTP in PBB network



AS-1 Configuration

NOTE
The AS-2 configuration is similar to the AS-1 configuration.

To carry PBB traffic, configure a VPLS instance. The B-VLAN used here is 100. For PB traffic, the S-VLAN used is 200 and C-VLAN 300.

Tag type configuration

```
device_AS-1(config)#tag-type 88e8 eth 1/1
device_AS-1(config)#tag-type 88e8 eth 1/2
device_AS-1(config)#tag-type 9100 eth 1/10
```

B-VLAN Configuration

```
device_AS-1(config)#vlan 100
device_AS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-vlan-100)#rstp
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pbb-bvlan 1
device_AS-1(config-mpls-vpls-pbb-bvlan)#pbb
device_AS-1(config-mpls-vpls-pbb-bvlan-pbb)#vlan 100 isid 101010
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tagged ethernet 1/1 ethernet 1/2
```

S-VLAN Configuration

```
device_AS-1(config-mpls-vpls-pbb-bvlan)#vlan 200
device_AS-1(config-mpls-vpls-pbb-bvlan-vlan-200)#tagged ethernet 1/10
```

CS-1 Configuration

NOTE
The configuration of CS-2 is similar to the configuration of CS-1, except for the RSTP priority configuration.

Port type configuration

```
device_CS-1(config)#tag-type 88e8 eth 1/1
device_CS-1(config)#tag-type 88e8 eth 1/2
```

B-VLAN Configuration

```
device_CS-1(config)#vlan 100
device_CS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

RSTP Configuration

```
device_CS-1(config-vlan-100)#rstp
device_CS-1(config-vlan-100)#rstp priority 100
```

ES-1 Configuration

NOTE

The configuration of ES-2 is similar to the configuration of ES-1.

Port type configuration

```
device_ES-1(config)#tag-type 9100 eth 1/10
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-svlan 1
device_ES-1(config-mpls-vpls-pb-svlan)#vlan 200
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/10
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-svlan#rstp
```

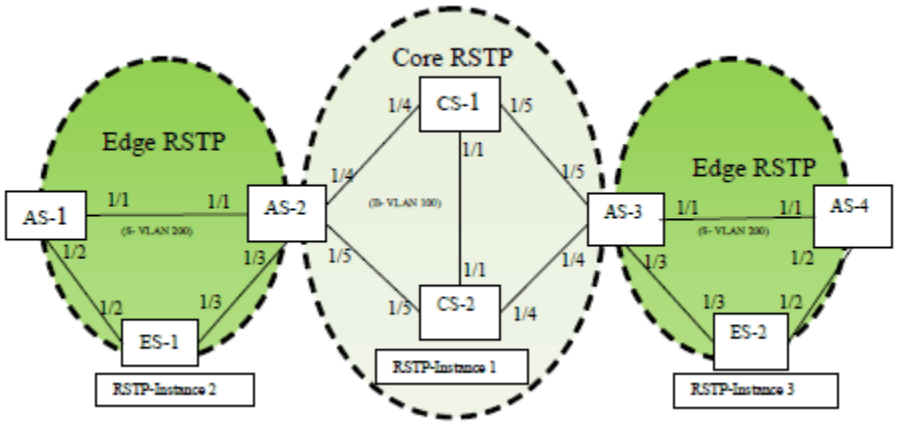
Use case 7: Core and Edge RSTP

RSTP will run in the PBB core for loop detection and avoidance in the B-VLAN. All ASs and CSs in B-VLAN will participate in the same RSTP instance. One CS will act as root for the core RSTP instance. All AS will have a different RSTP instance on the ES facing side and one of the AS will be acting as a ROOT.

The following deployment scenario is a case where RSTP is deployed for a B-VLAN and S-VLAN in a PBB network. In CS-1 and CS-2 a regular VLAN 100 is configured as a B-VLAN. AS-1 and AS-2 has a VPLS VLAN 100 which acts as a B-VLAN. VPLS VLAN 200 on ES-1 and ES-2 which acts as an S-VLAN for the PB network. CS-1 is configured as the ROOT Bridge for core RSTP. AS-1 and AS-4 are the ROOT bridges for RSTP on the ES facing side.

The following discussion describes how to configure the nodes in the topology.

FIGURE 107 RSTP deployment in PB and PBB network



Configuring AS-1

NOTE
The configuration of AS-4 is similar to the configuration of AS-1.

Tag type configuration

Configure aport tag type for S-VLAN to 0x9100.

```
device_AS-1(config)#tag-type 9100 eth 1/1
device_AS-1(config)#tag-type 9100 eth 1/2
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-1(config)#router mpls
device_AS-1(config-mpls)#vpls pb-svlan 1
device_AS-1(config-mpls-vpls-pb-svlan)#pbb
device_AS-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/2
```

RSTP Configuration

```
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#rstp
device_AS-1(config-mpls-vpls-pb-svlan-vlan-200)#rstp priority 100
```

Configuring AS-2

NOTE
The configuration of AS-3 is similar to the configuration of AS-2.

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_AS-2(config)#tag-type 9100 eth 1/1
device_AS-2(config)#tag-type 9100 eth 1/3
device_AS-2(config)#tag-type 88e8 eth 1/5
device_AS-2(config)#tag-type 88e8 eth 1/6
```

B-VLAN configuration

```
device_AS-1(config)#vlan 100
device_AS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/2
```

RSTP configuration

```
device_AS-1(config-vlan-100)#rstp
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_AS-2(config)#router mpls
device_AS-2(config-mpls)#vpls pb-svlan 1
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/1 ethernet 1/3
device_AS-2(config-mpls-vpls-pb-svlan-vlan-100)#rstp
```

B-VLAN Configuration

```
device_AS-2(config-mpls-vpls-pb-svlan)#pbb
device_AS-2(config-mpls-vpls-pbb-bvlan-pbb)#vlan 100 isid 101010
device_AS-2(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#tag ethernet 1/4 ethernet 1/5
```

RSTP Configuration

```
device_AS-2(config-mpls-vpls-pbb-bvlan-vlan-100-isid-101010)#rstp
device_AS-2(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_AS-2(config-mpls-vpls-pb-svlan-vlan-200)#rstp
```

Configuring ES-1

NOTE

The configuration of ES-2 is similar to the configuration of ES-1.

Tag type configuration

Configure port tag type for S-VLAN to 0x9100.

```
device_ES-1(config)#tag-type 9100 eth 1/2
device_ES-1(config)#tag-type 9100 eth 1/3
```

S-VLAN Configuration

Configure VPLS VLANs which will carry the PB traffic, the S-VLAN here is 200.

```
device_ES-1(config)#router mpls
device_ES-1(config-mpls)#vpls pb-svlan 1
device_ES-1(config-mpls-vpls-pb-svlan)#pbb
device_ES-1(config-mpls-vpls-pb-svlan-pbb)#vlan 200
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#tag ethernet 1/2 ethernet 1/3
```

RSTP Configuration

```
device_ES-1(config-mpls-vpls-pb-svlan-vlan-200)#rstp
```

CS-1 Configuration:

NOTE

The configuration of CS-2 is similar to the configuration of CS-1, except for the RSTP priority configuration.

Port type configuration

```
device_CS-1(config)#tag-type 88e8 eth 1/1
device_CS-1(config)#tag-type 88e8 eth 1/2
device_CS-1(config)#tag-type 88e8 eth 1/2
```

B-VLAN Configuration

```
device_CS-1(config)#vlan 100
device_CS-1(config-vlan-100)#tagged ethernet 1/1 ethernet 1/4  ethernet 1/5
```

RSTP Configuration

```
device_CS-1(config-vlan-100)#rstp
device_CS-1(config-vlan-100)#rstp priority 100
```

NOTE

There can be scenarios where the S-VLAN used on ES-1 is different from one used on ES-2. This requires the configuration of different S-VLANs on AS-1 and AS-2.

Metro Ring Protocol

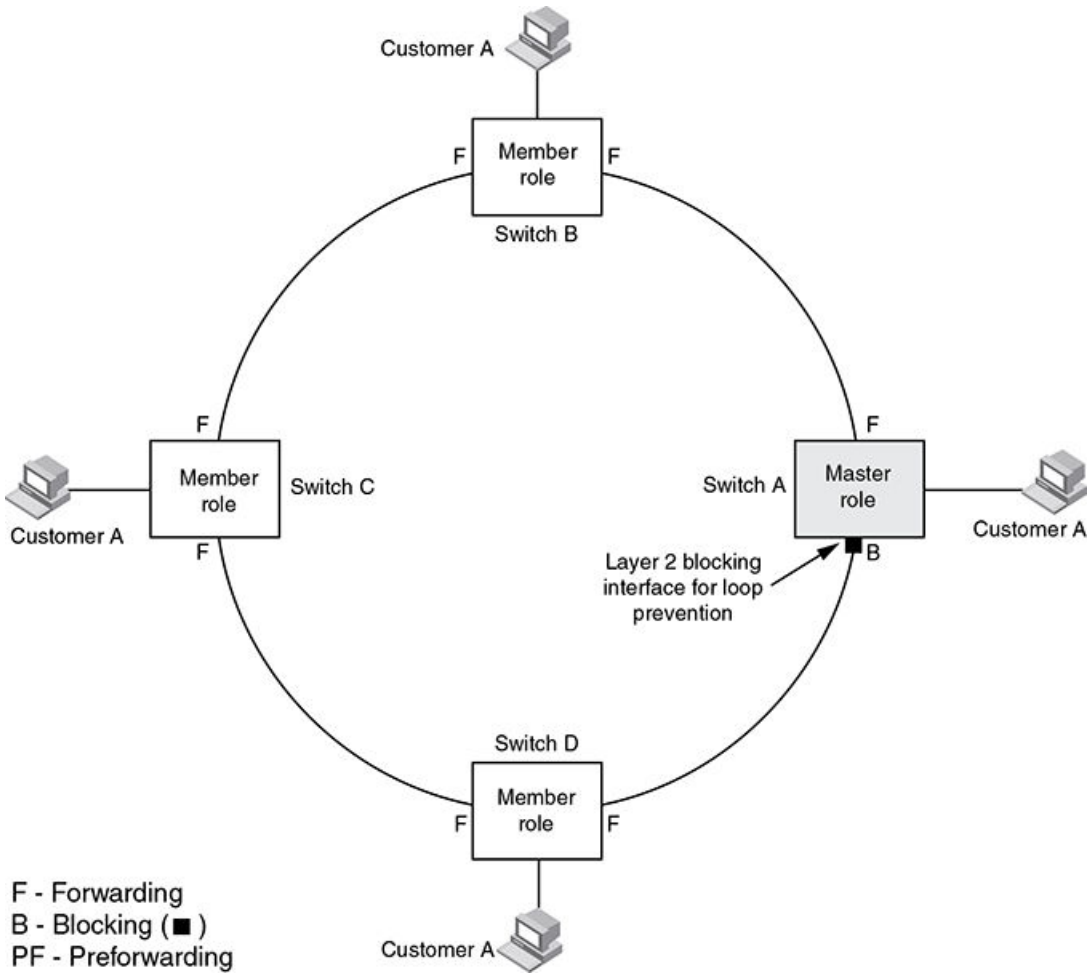
Metro Ring Protocol

Brocade Metro Ring protocol (MRP) is a proprietary protocol that prevents layer 2 loops and provides fast reconvergence in ring topologies. It is an alternative to Spanning Tree Protocol (STP) and is especially useful in Metropolitan Area Networks (MAN) where using 802.1D STP has the following drawbacks:

- 802.1D recommends a maximum bridge diameter of seven nodes with standard timers. MRP is capable of many more nodes than this.
- 802.1D has a slow reconvergence time, taking many seconds or even minutes. MRP can detect and heal a break in the ring in under one second.

Figure 110 shows a simple metro ring.

FIGURE 108 MRP - normal state



The ring in this example consists of four Brocade device nodes that support MRP. Each node has two ring interfaces and the interfaces are all in one port-based vlan. There are customer networks utilizing the nodes and layer 2 traffic is forwarded to and from the customer networks through the ring. Each customer interface can be in the same vlan as the ring or in a separate vlan under control of MRP as part of a topology group.

For each discrete ring one node is configured in the master role for the MRP ring. One of the two ring interfaces on the master node is configured as the primary interface, the other is the secondary interface. The primary interface originates Ring Health Packets (RHPs) which are used to monitor the health of the ring. An RHP is forwarded on the ring to the next interface until it reaches the secondary interface of the master node. On receipt of an RHP the secondary interface transitions into blocking mode to prevent a layer 2 loop.

Table 61 displays the individual Brocade devices and the MRP features they support.

TABLE 61 Supported Brocade MRP features

Features supported	Brocade NetIron XMR Series	Brocade NetIron MLX Series	Brocade NetIron CES Series 2000 Series BASE package	Brocade NetIron CES Series 2000 Series ME_PREM package	Brocade NetIron CES Series 2000 Series L3_PREM package	Brocade NetIron CER Series 2000 Series Base package	Brocade NetIron CER Series 2000 Series Advanced Services package
MRP Phase 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MRP Phase 2	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MRP Alarm RHP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MRP and MRP-II support under an ESI with support for B-VLANs, S-VLANs and C-VLANs	No	No	No	Yes	No	No	Yes
MRP Diagnostics	Yes	Yes	Yes	Yes	Yes	Yes	Yes

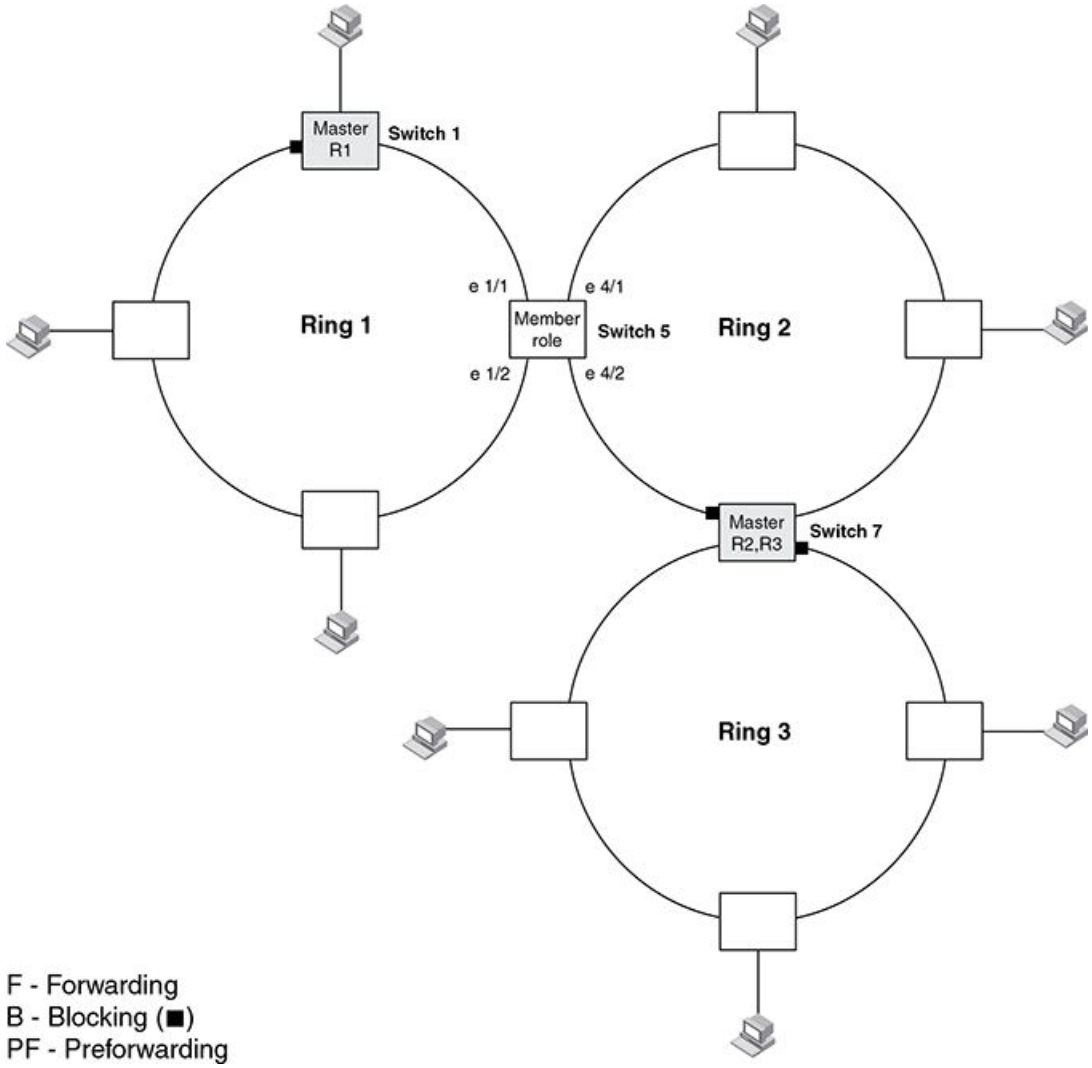
NOTE

When you configure MRP, it is recommended that you disable the secondary ring interface on the master node before beginning or changing the ring configuration. Disabling an interface prevents a layer 2 loop from occurring while you are configuring MRP on the ring nodes. Once you have completed the MRP configuration and enabled it on all the nodes, you should re-enable the secondary ring interface.

MRP rings without shared interfaces (MRP Phase 1)

MRP Phase 1 allows you to configure multiple MRP rings, as shown in Figure 111, but the rings cannot share the same interfaces. For example, you cannot configure ring 1 and ring 2 to share interfaces ethernet 1/1 and 1/2 on Switch 5. Each ring must remain an independent ring and RHP packets are processed within each ring.

FIGURE 109 MRP - multiple rings, no shared interfaces

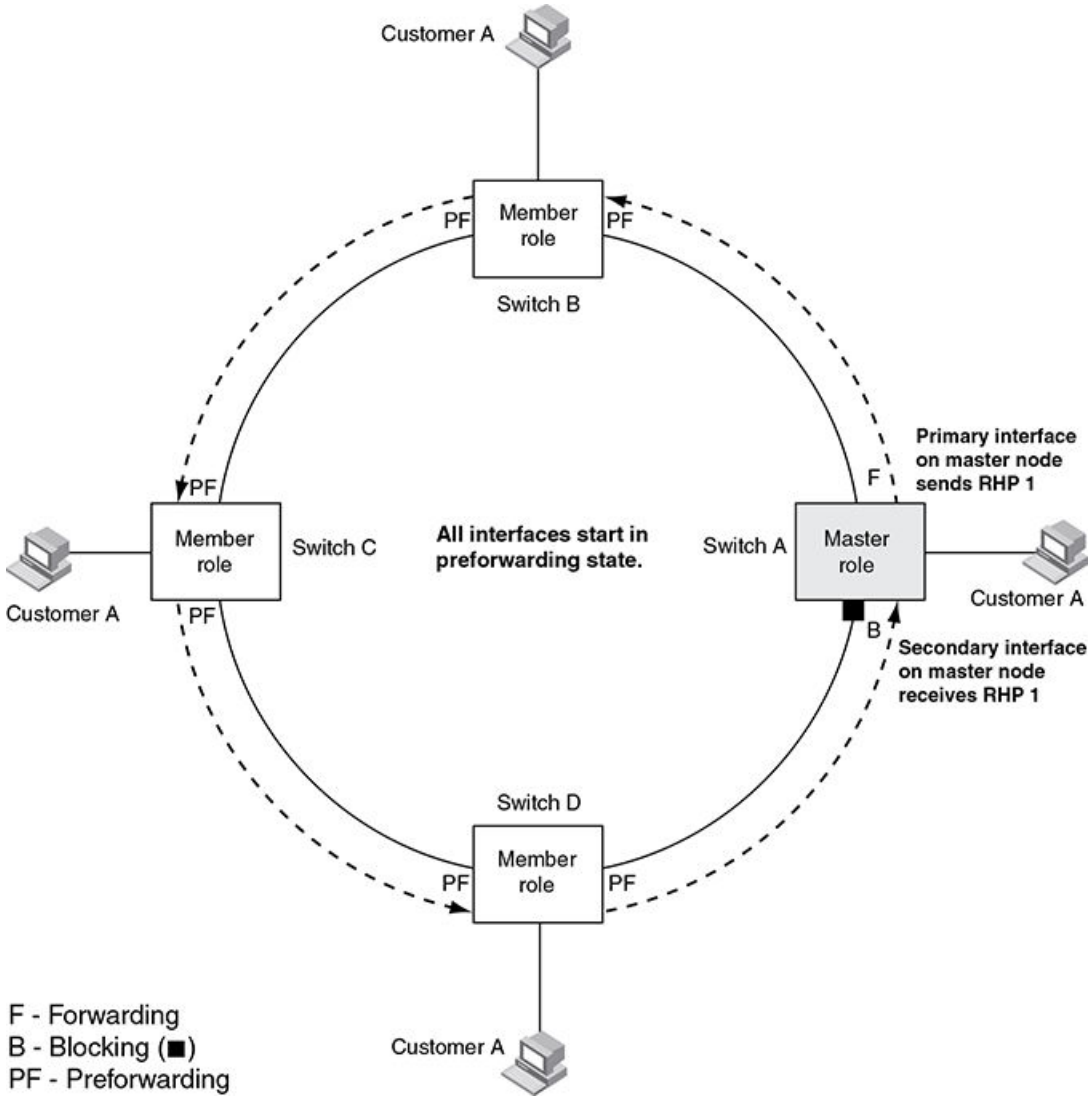


In the above example, Switch 5 and Switch 7 are configured with multiple MRP rings however each ring has discrete ring interfaces allocated to it to prevent any sharing. Any ring node can be the master for its ring and a node can be the master for more than one ring as shown on Switch 7 due to the separation of rings.

Ring initialization

Figure 112 shows the initial state of the ring, when MRP is first enabled on the ring's switches. On the master the primary interface starts in forwarding mode and the secondary interface starts in blocking mode. All ring interfaces on member nodes begin in the preforwarding state (PF).

FIGURE 110 MRP ring - initial state



An RHP is an MRP protocol packet used to monitor the health of the ring. The source address is the MAC address of the master node and the destination MAC address is a protocol address for MRP. The Master node generates RHPs and sends them on the ring. The state of a ring interface is influenced by the RHPs.

A ring interface can have one of the following MRP states:

- **Preforwarding (PF)** - The interface will forward RHPs and learn MAC addresses but won't forward data for the ring. All ring interfaces start in this state when you enable MRP except the master node. A blocking interface transitions to preforwarding when the preforwarding timer expires and no RHP's have been received.
- **Forwarding (F)** - The interface will forward RHP's and data for the ring. On member switches an interface transitions from preforwarding to forwarding when the preforwarding time expires or the interface receives an RHP with the forwarding bit set. A break in the ring is indicated if the secondary interface on the master fails to receive an RHP within the preforwarding timer and the interface transitions from blocking to forwarding to heal the ring.

- **Blocking (B)** - The interface can process RHPs, but cannot forward data for the ring. Only the secondary interface on the master node can be blocking.

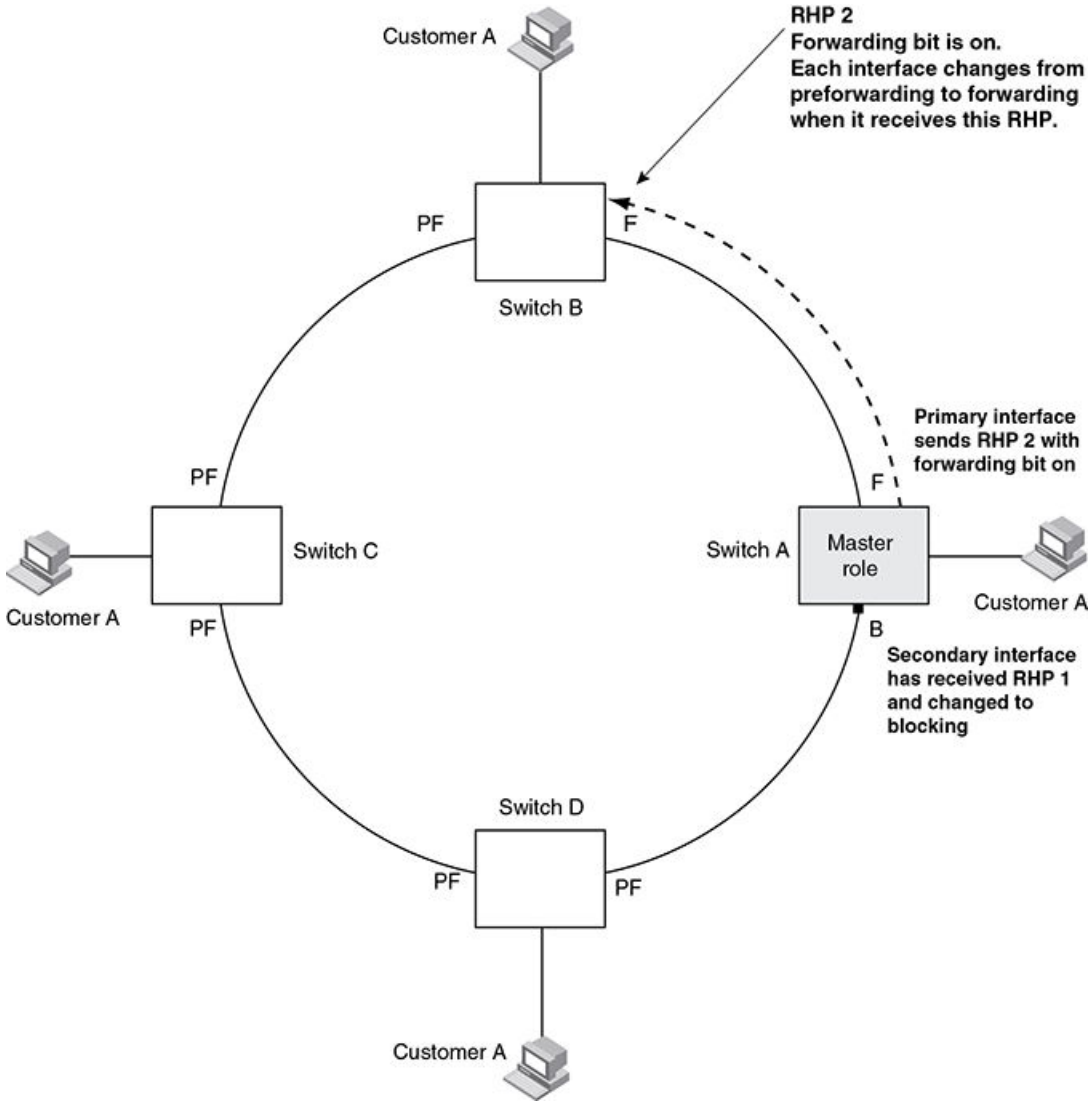
  NOTE
  The configured preforwarding time defines the number of milliseconds the interface will remain in a state before changing to the next state without receiving an RHP.

When MRP is enabled, all interfaces begin in the preforwarding state and the primary interface on the master node immediately sends an RHP (RHP 1 in Figure 112) onto the ring. The secondary interface on the master node listens for the RHP:

- If the secondary interface receives the RHP, all links in the ring are up and the interface changes its state to blocking. The primary interface then sends another RHP (RHP 2 in Figure 113) with its forwarding bit set on. As each of the member interfaces receives the RHP, the interfaces change their state to forwarding. Typically, this occurs in sub-second time. The ring very quickly enters the fully initialized state.
- If the secondary interface does not receive the RHP by the time the preforwarding time expires, a break has occurred in the ring. The secondary interface changes its state to forwarding. The ring is not intact, but data is still forwarded among the nodes using the links that are up.

Figure 113 shows an example.

FIGURE 111 MRP ring - from preforwarding to forwarding

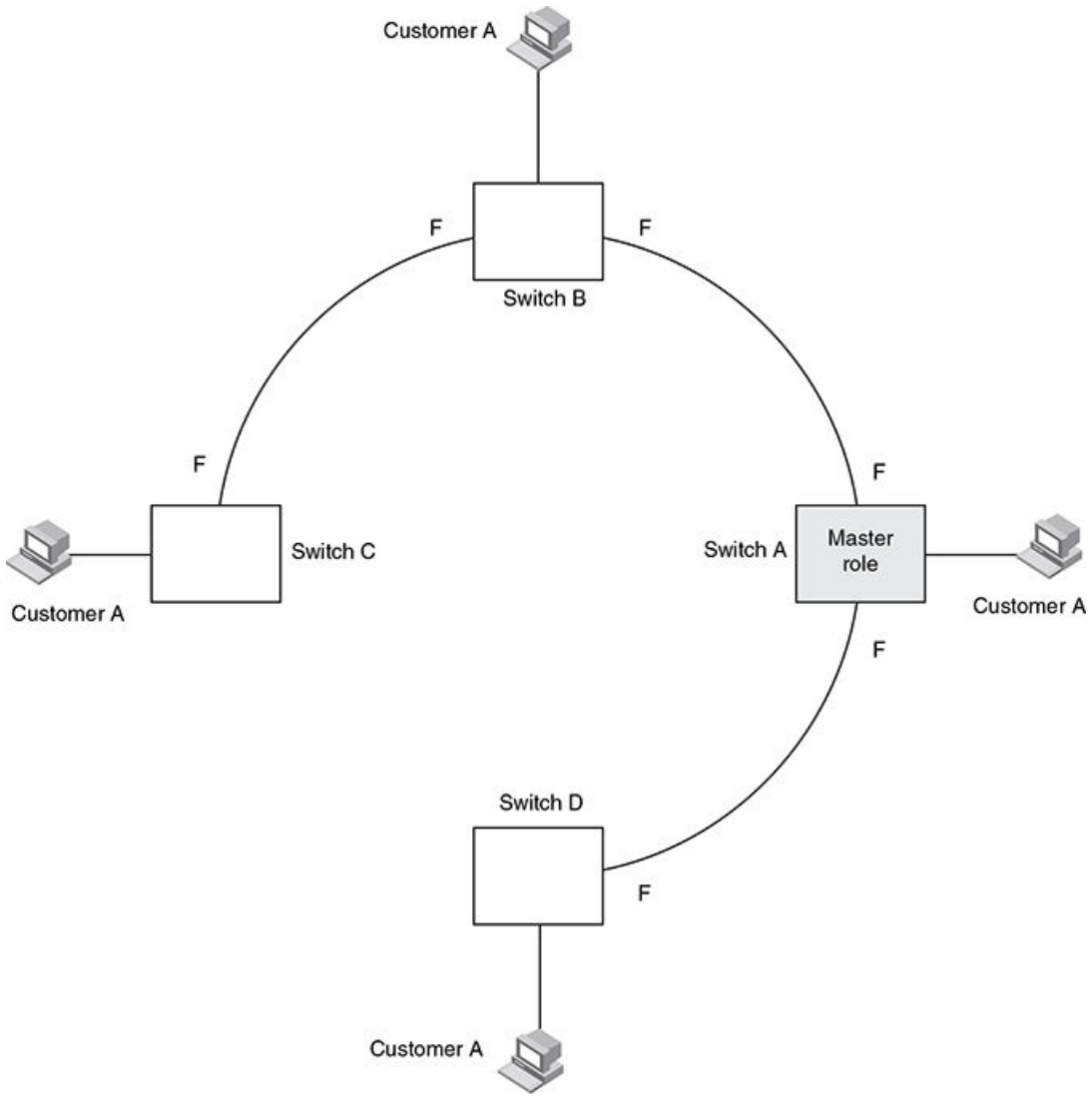


Each RHP also has a sequence number. MRP can use the sequence number to determine the round-trip time for RHPs in the ring. Refer to Using MRP diagnostics on page 480.

How ring breaks are detected and healed

Figure 114 shows ring interface states following a link break. MRP quickly heals the ring and preserves connectivity among the customer networks.

FIGURE 112 MRP ring - ring break



If a break in the ring occurs, MRP heals the ring by changing the states of some of the ring interfaces:

- **Blocking interface** - When the secondary interface on the master node transitions to a blocking state it sets a timer defined by the preforwarding time configured. If the timer expires before the interface receives a ring RHP, the interface changes state to preforwarding. Once the secondary interface state is preforwarding:
  - If the interface receives an RHP, the interface changes back to the blocking state and resets the timer.
  - If the interface does not receive an RHP for its ring before the preforwarding time expires, the interface changes to the forwarding state, as shown in Figure 114.
- **Forwarding interfaces** - All member interfaces remain in the forwarding state unless the physical interface is in an error condition.

When the link is repaired, the associated MRP interfaces come up in the preforwarding state allowing RHPs to be forwarded around the ring and finally reach the secondary interface on the master node:

- If an RHP reaches the master node's secondary interface, the ring is intact, the secondary interface changes to blocking. The master node sets the forwarding bit on in the next RHP. When the restored interfaces receive this RHP, they immediately change state to forwarding.
- If an RHP does not reach the master node's secondary interface, the ring is still broken. The master node does not send an RHP with the forwarding bit on. In this case, the restored interfaces remain in the preforwarding state until the preforwarding timer expires, then change to the forwarding state.

MRP alarm RHP enhancement

Prior to the enhancement detection of ring breaks were completely timer based. If the ring master fails to receive RHPs for a period of 3 "hello times" (by default the hello time is 100 ms) this indicates that the ring is broken in some manner. This initiates a topology change as described in the previous section. The convergence time associated with such an event could take several hundred milliseconds.

This enhancement enables ring nodes to rapidly notify the master of link failures. To understand the mechanism we introduce the concept of downstream switches in the ring and how member switches determine the primary and secondary ring interfaces. Remember that a primary ring interface sends RHPs and a secondary ring interface receives RHPs.

To fully understand the mechanism the reader needs to be aware of the concept of shared interfaces and interface owner ID's which are a function of MRP phase 2.

A downstream switch is defined as the next switch that will receive the ring RHP originated from the master primary interface for a particular ring. In Figure 115 Switch B is downstream from the master, Switch C is downstream from Switch B and so on and so forth. In addition it should be noted that a member switch identifies which ring interface is secondary for each discrete ring by virtue of the receipt of RHPs for that ring. In a topology with shared interfaces a single physical interface can therefore be a primary ring interface for one ring and a secondary ring interface for another ring. It should be noted that the output of the 'show metro' command as well as the configuration will change if the primary and secondary ring interfaces of the master are swapped. This keeps the identification of interface roles consistent with the flow of RHPs for discrete ring instances.

When a link is detected to be down on a member switch secondary ring interface due to a link failure an alarm RHP, which is an RHP with the alarm bit set, is sent from the primary ring interface towards the ring master, notifying the master of the failure.

The destination MAC address in the alarm is the ring MAC address. The MAC address will be in the format 0304.8000.00xx where 'xx' is the ring number in hexadecimal.

For example ring 100 = 0304.8000.0064. This ensures that the packet is hardware forwarded all the way to the master. When the master in the ring receives this alarm the secondary interface is immediately transitioned from blocking to forwarding.

NOTE

In the event of a shared interface failing the alarm RHP packet is only sent by the owner ring of the failed interface. If all rings configured on a shared interface were to generate alarms then the respective master switches for each ring would start forwarding on both interfaces creating a loop condition. By restricting alarm generation to the owner ring we ensure that only one master switch is notified to ensure that the ring heals. The owner ring ID should be the highest priority ring configured on the shared interface.

Operation of the alarm RHP enhancement is shown in Figure 115 and described below:

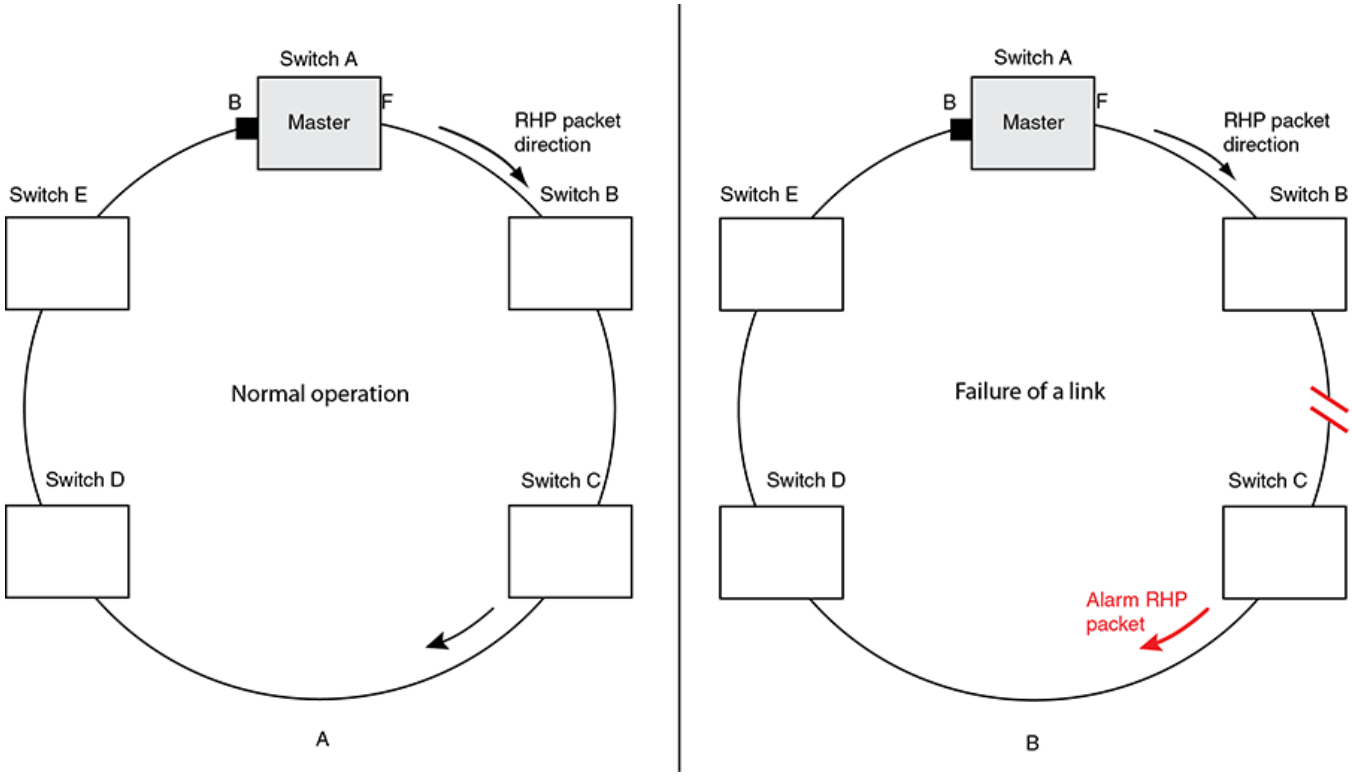
When the link between Switch B and Switch C fails, the downstream switch detects the failure of the link associated with its secondary ring interface and generates an alarm. The following is the complete sequence of events that occurs.

1. The downstream Switch C detects a link down event on the link to its upstream neighbor Switch B.
2. Switch C sends a single RHP packet with the alarm bit set. The RHP packet is sent in the same direction of flow as that of the normal RHP packets.

3. Switch A receives the alarm on the secondary ring interface that was sent by Switch C. It is now aware that the ring is broken even though the preforwarding timer for blocking to preforwarding may not have expired.
4. Switch A immediately transitions its secondary interface from blocking to forwarding to heal the ring.
5. RHP packets continue to be sent on the primary interface by Switch A to detect when the ring has been healed.

   From a user perspective there is no other difference in the behavior of the ring other than the rapid convergence due to link failures. There is no CLI command required to enable this feature.

FIGURE 113 An MRP ring under normal operation (A) and after detection of a failure in the ring (B)



Topology change notification for multicast traffic

Figure 116 shows a Layer 2 aggregation network which runs on Brocade MRP. In this scenario, switch A acts as the MRP master and also the Internet Group Management Protocol (IGMP) querier. When a link failure is detected on the port 1/1 of switch A, the port 1/2 of switch A will transition from blocking state to forwarding state allowing the ring to re-converge in sub-second time. With the transition of port 1/2 to forwarding state, the IGMP querier sends the IGMP reports through the port 1/2 and the IGMP receivers will send the join message to the querier causing the multicast traffic to re-converge immediately.

FIGURE 114 MRP ring with switch A as the querier

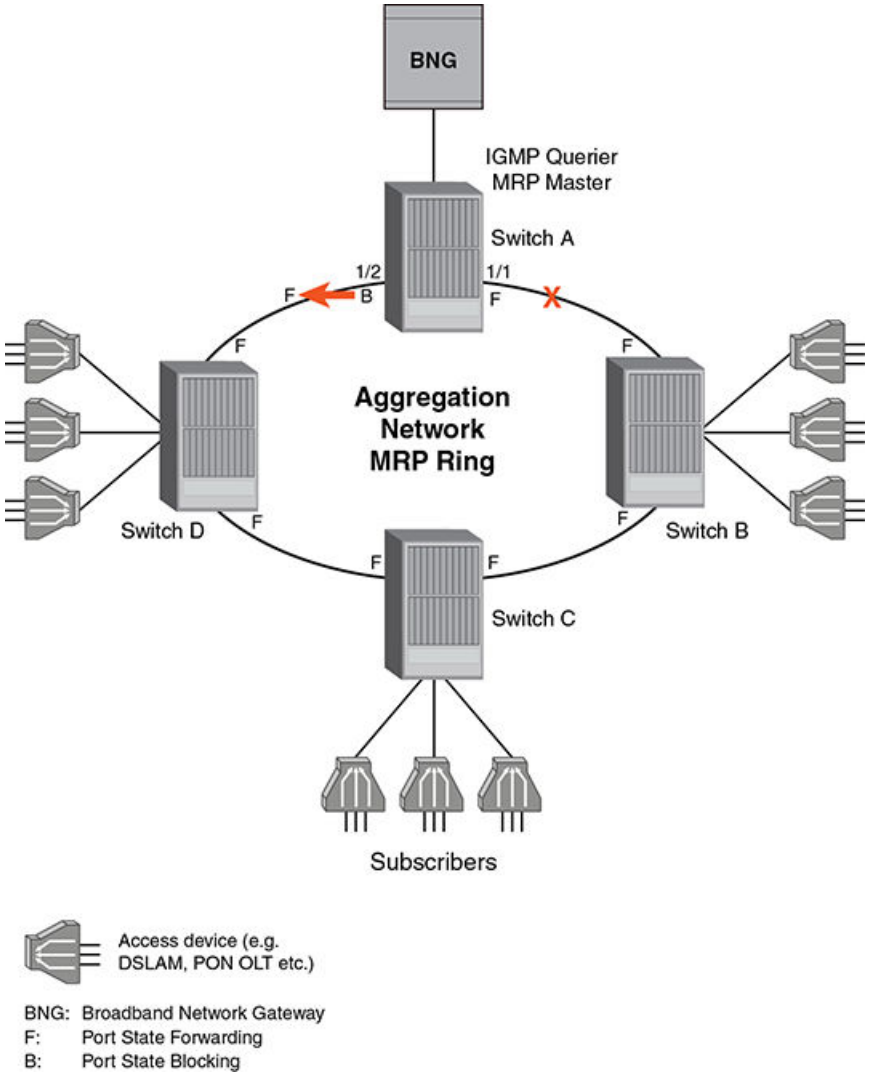
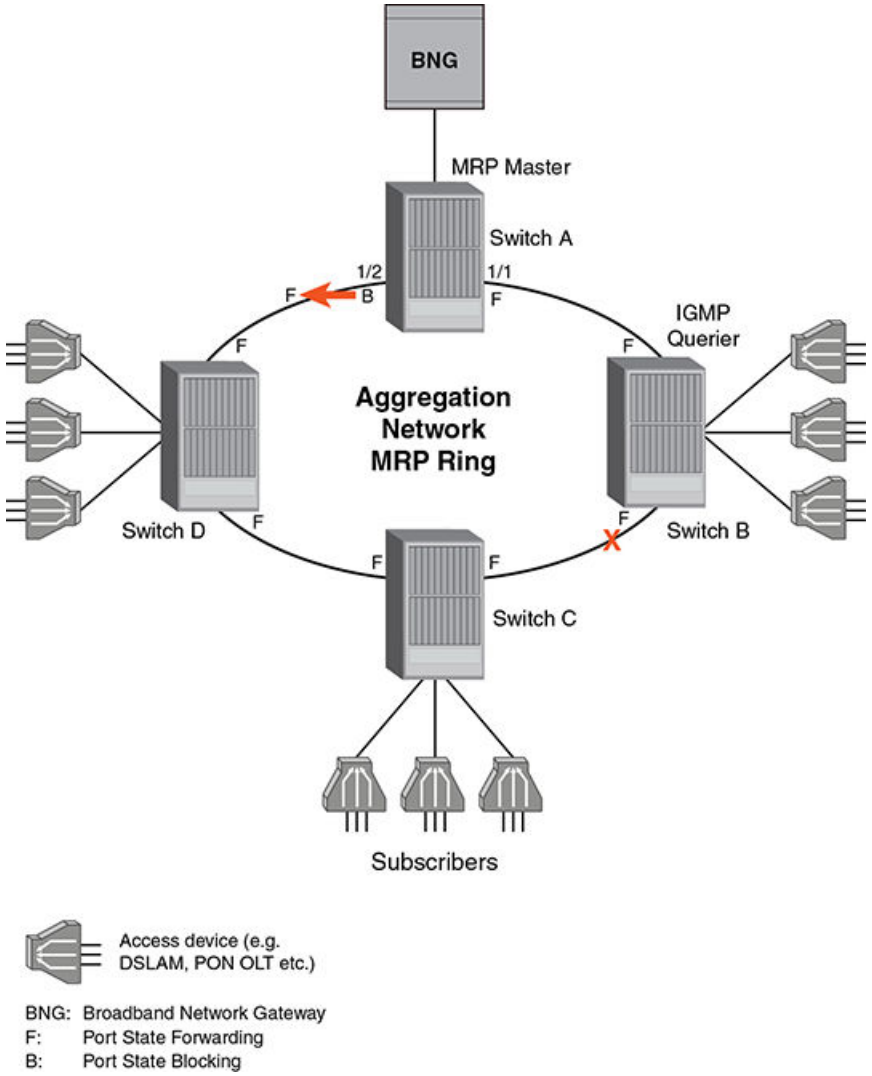


Figure 117 shows a Layer 2 aggregation network which runs on Brocade MRP. In this scenario, switch A acts as the MRP master and switch B acts as the IGMP querier. When a link failure is detected on the switch B, the port 1/2 of switch A will transition from blocking state to forwarding state causing the multicast traffic to re-converge. The failover time of the multicast traffic is determined by the IGMP query interval and the IGMP group membership time on the querier. These timers can be set by **ip igmp query-interval** and **ip igmp group-membership-time** commands.

FIGURE 115 MRP ring with switch B as the querier



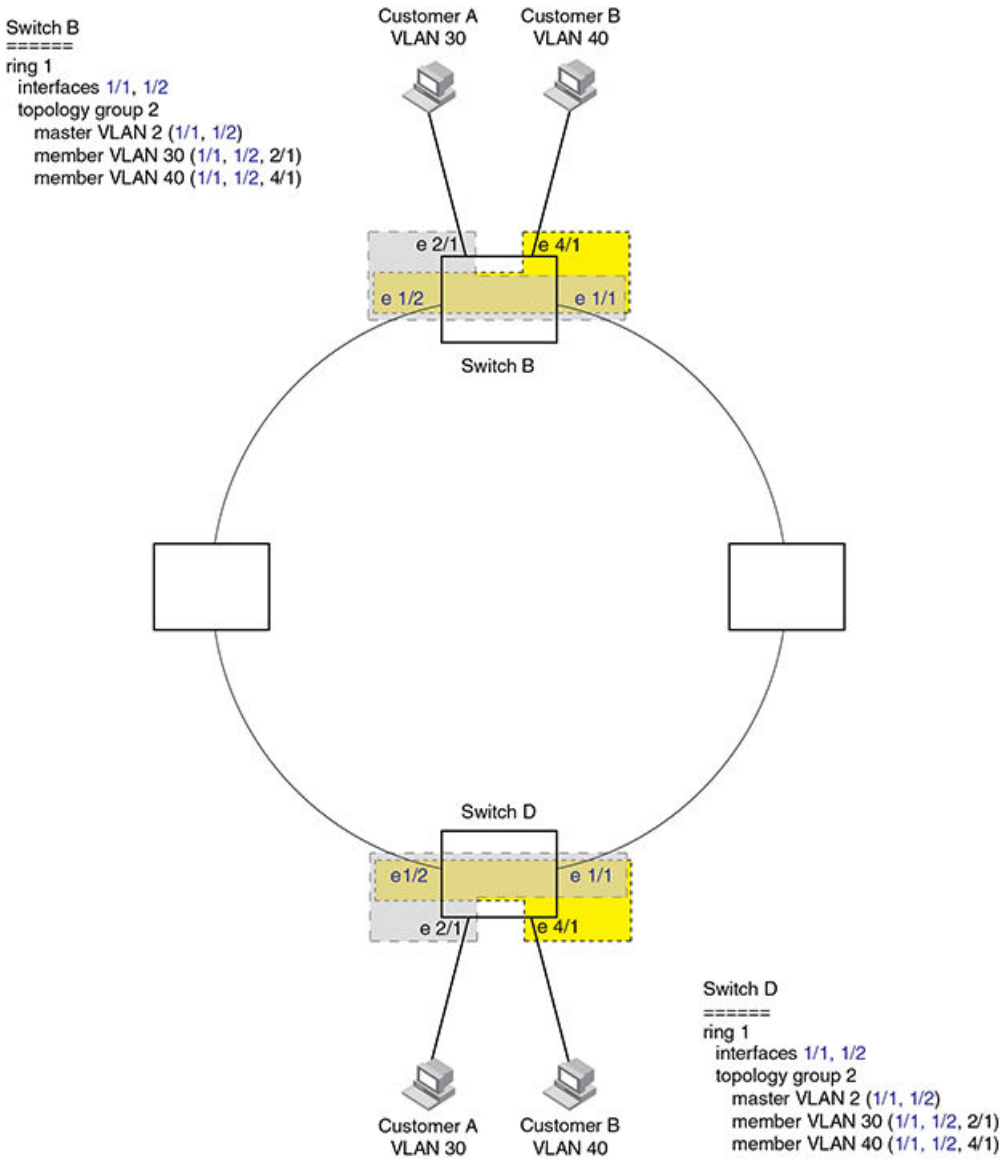
In the scenarios shown in Figure 116 and Figure 117, a topology change notification is sent to the MRP ring master, which forces an advertisement of the IGMP query to the entire network upon receiving the notification. This reduces the failover time for the multicast streams over a ring topology without the need of lowering the IGMP query interval or IGMP group membership time.

Master VLANs and member VLANs in a topology group

The reader is referred to *Topology Groups* chapter for further information on topology group concepts and operation.

All the ring interfaces must be placed into the master vlan for the topology group. Customers configured with member vlans inherit the configuration of the topology group master vlan and have equivalent layer 2 connectivity across the ring. Figure 118 shows an example.

FIGURE 116 MRP ring - ring vlan and customer vlan



In this example each customer has their own vlan. Customer A has vlan 30 and customer B has vlan 40.

Customer A host attached to Switch D on an interface in vlan 30 can reach the customer A host attached to Switch B on an interface in vlan 30 through the ring at layer 2. The same mechanism is used to connect customer B hosts on vlan 40.

Customer A and customer B traffic is separated by using different vlans.

You can configure MRP separately on each customer vlan. However, this is impractical if you have many customers. To simplify configuration when you have a lot of customers (and therefore a lot of vlans), you can use a topology group.

A topology group enables you to control forwarding in multiple vlans using a single instance of a layer 2 protocol such as MRP. A topology group contains a master vlan and member vlans. The master vlan contains all the configuration parameters for the layer 2 protocol (STP, MRP, or VSRP). The member vlans use the layer 2 configuration of the master vlan.

In Figure 118, vlan 2 is the master vlan and contains the MRP configuration parameters for ring 1. vlan 30 and vlan 40, the customer vlans, are member vlans in the topology group. Since a topology group is used, a single instance of MRP provides redundancy and loop prevention for both the customer vlans.

If you use a topology group:

- The master vlan must contain the ring interfaces.
- The ring interfaces must be tagged as they will be used for multiple vlans.
- The member vlan for a customer must contain the two ring interfaces and the interfaces for the customer.

Refer to MRP CLI example on page 483 for the configuration commands required to implement the MRP configuration shown in Figure 118.

Configuring MRP

To configure MRP, perform the following tasks for each discrete ring:

- On the switch identified as the ring master disable the secondary ring interface. This manually prevents a layer 2 loop from occurring during configuration.
- Configure each switch for MRP one at a time following the planned flow of RHP's
- On each switch in the path add an MRP ring to a port-based vlan. When you add a ring, the CLI changes to the configuration level for the ring, where you can do the following:
  - On the master node configure the master ring role.
  - Specify the two MRP interfaces for the ring
  - Option: Specify a name for the ring. Brocade recommends that you have a naming convention for your MRP rings and consistently apply names for all the rings in the topology.
  - Option: Change the hello time and the preforwarding time. These parameters control how quickly failover occurs if the master fails to receive RHPs for the ring.
  - Enable the ring.
- Re-enable the interface you disabled in step one. MRP will prevent loops when enabled on all devices in the ring.

When using topology groups the ring configuration must be added to the master-vlan for the group. For further information refer to *Topology Groups* chapter.

Adding an MRP ring to a vlan

> NOTE
> If you plan to use a topology group make sure you configure MRP on the topology group's master vlan.

To add a MRP ring to a vlan, enter commands such as the following.

```
device(config)# vlan 2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name CustomerA
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
```

These commands configure an MRP ring in vlan 2 with a ring ID of 1, a ring name of Customer A. If the node is the master then the following command is used to specify the node as the master for the ring.

device(config-vlan-2-mrp-1)# master

The ring interfaces are 1/1 and 1/2. The first interface listed will be allocated as the primary interface and the second will be allocated as the secondary interface. The primary interface initiates RHPs. The ring takes effect in vlan 2.

Syntax: [no] metro-ring ring-id

The *ring-id* parameter specifies the ring ID 1 - 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name string

The *string* parameter specifies a name for the ring. The name is optional, but it can be up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] master

Configures this node as the master node for the ring. Enter this command only on one node in the ring. The node is a member (non-master) node by default.

Syntax: [no] ring-interface ethernet primary-if ethernet secondary-if

The **ethernet** *primary-if* parameter specifies the primary interface. On the master node, the primary interface originates RHPs. Ring control traffic will flow out of this interface by default. On member nodes the order in which you enter the interfaces does not matter as the secondary interface is determined by the receipt of RHP's from the master meaning the other interface defined in config becomes the primary. Once the ring is enabled the configuration entries on a member switch will reflect the ring direction no matter what order they are originally entered.

The **ethernet** *secondary-if* parameter specifies the secondary interface.

> NOTE
> The Brocade NetIron CES Series and Brocade NetIron CER Series devices do not support selection of a secondary interface based on reception of RHPs. As a result, the primary and secondary interfaces must be configured correctly.

Syntax: [no] enable

The **enable** command enables the ring.

Changing the hello and preforwarding times

You can also change the RHP hello time and preforwarding time. To do so, enter commands such as the following.

```
device(config-vlan-2-mrp-1)# hello-time 200
device(config-vlan-2-mrp-1)# preforwarding-time 400
```

These commands change the hello time to 200 ms and change the preforwarding time to 400 ms.

Syntax: [no] hello-time ms

Syntax: [no] preforwarding-time ms

The *ms* specifies the number of milliseconds.

The hello time can be from 100 - 1000 (one second). The default hello time is 100 ms.

The preforwarding time can be from 200 - 5000 ms, and must be at least twice the value of the hello time and must be a multiple of the hello time. The default preforwarding time is 300 ms.

A change to the hello time or preforwarding time takes effect as soon as you enter the command.

> NOTE
> You can use MRP ring diagnostics to determine whether you need to change the hello time and preforwarding time. Refer to Using MRP diagnostics on page 480.

Changing the scale timer

You are able to decrease MRP convergence time by changing the MRP scale timer tick from 100 ms to 50 ms. To do so, enter the following command:

```
device(config)# scale-timer mrp
```

Syntax: [no] scale-timer mrp

Note: This command accepts no values and is put in place as it is shown above.

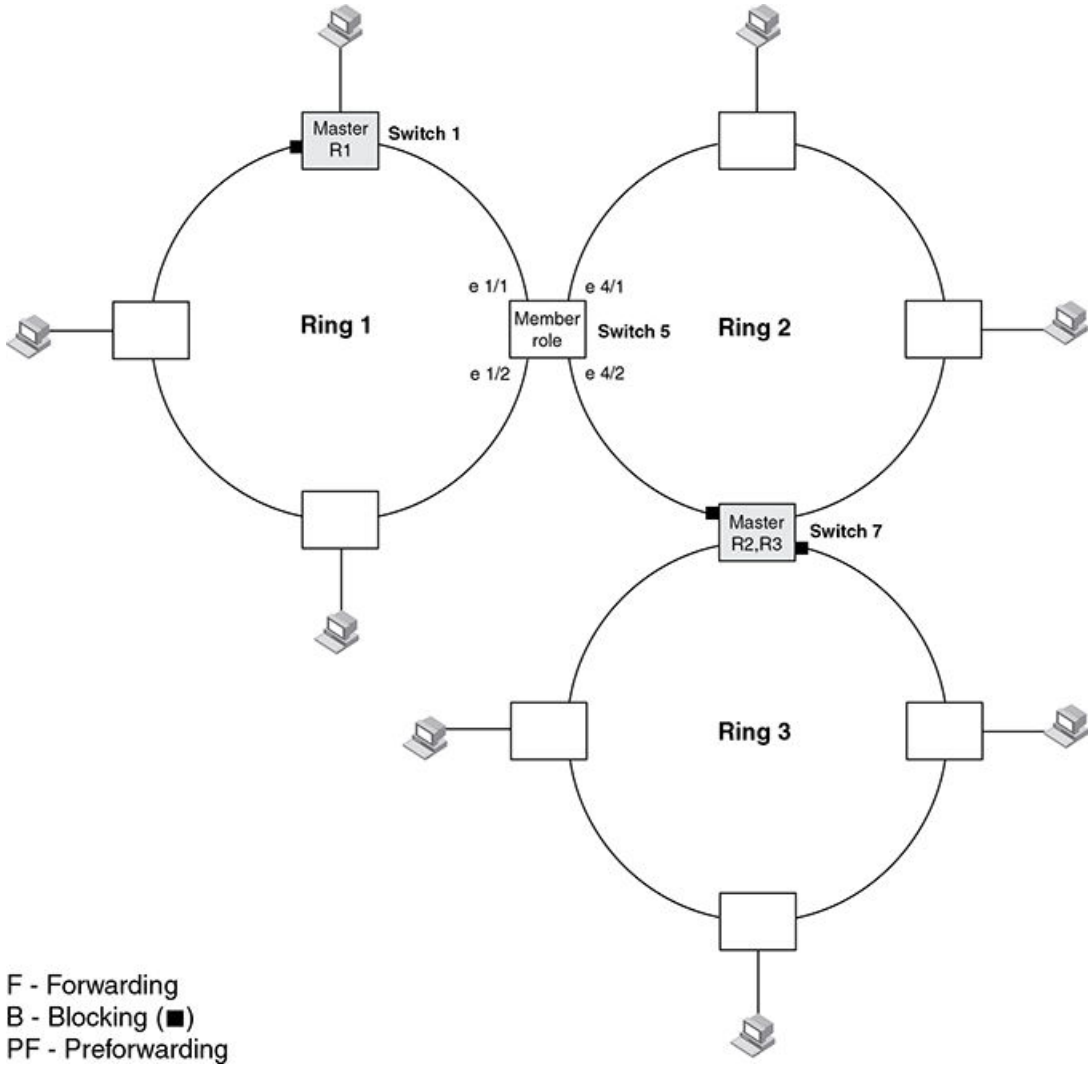
Note: Changing the scale timer affects the operation of MRP. Refer to Tuning MRP timers on page 478 for further information.

MRP Phase 2

MRP phase 2 expands functionality by allowing a physical interface to be shared by multiple rings configured within the same vlan.

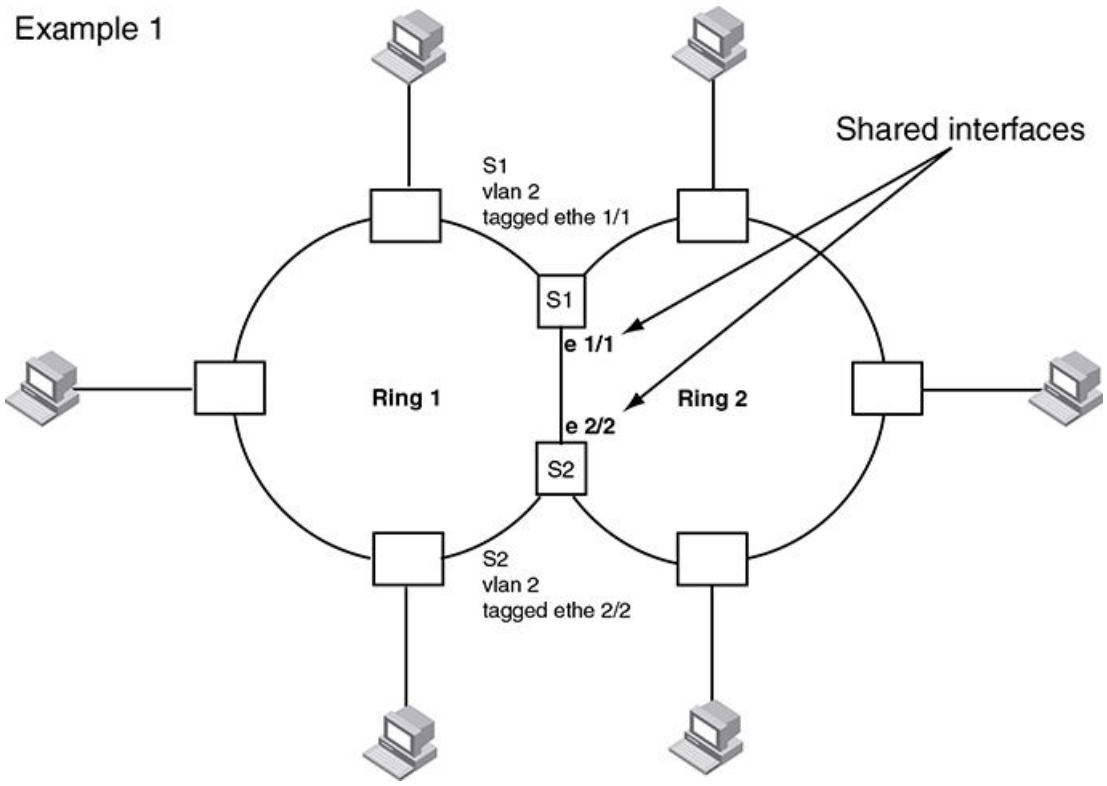
Recall that in MRP Phase 1, a node can have multiple MRP rings, but the rings cannot share the same interface. Any node can be designated as the master node for the ring. Each ring is an independent ring and RHP packets are processed within each ring exclusively.

FIGURE 117 Multiple MRP rings - phase 1



With MRP phase 2 multiple rings can be configured to share the same interface as long as the interfaces belong to the same vlan. Figure 120 shows an example of two rings that share the same interfaces on S1 and S2.

FIGURE 118 Example 1 multiple rings sharing interfaces - phase 2



On each node that will participate in ring 1, you configure the ring ID and the ring interfaces that will be used. You repeat the configuration steps for all nodes in ring 2. In a multiple ring configuration, a ring's ID determines its priority. The lower the ring ID, the higher the priority of a ring with ring ID 1 being the highest possible priority.

A key concept with MRP phase 2 is the ability to extend a single vlan across the whole topology even when multiple rings are required. Consider the example in Figure 121 where we have 3 MRP rings and a customer who needs to create neighbor relationships between all three routers depicted. The routers all have interfaces configured in a single subnet and need IP connectivity between each other.

If each ring had an independent vlan then we would have to have a mechanism to move IP packets from a single IP subnet between different layer 2 topologies. By using MRP phase 2 we have multiple rings all associated with a single layer 2 topology allowing a common subnet to be distributed in the manner shown in the example. Whilst this looks nothing like a standard spanning tree network it should be treated in the same way from the perspective of a layer 2 topology, multiple paths where certain paths must be blocked to prevent loops at layer 2.

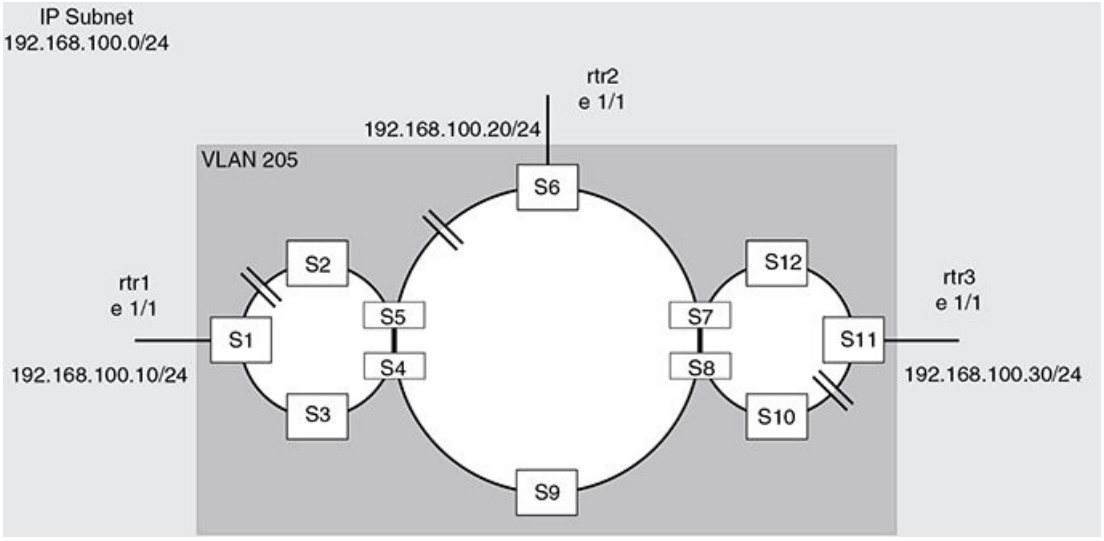
In addition it should be noted that the concept of multiple rings being associated with a single vlan describes the extent to which broadcasts will be propagated at layer 2 for that vlan. In other words a broadcast will be propagated to all ring interfaces in the layer 2 topology. The use of topology groups allows multiple vlans to effectively reuse a single layer 2 topology while maintaining a level of separation.

The obvious issue with this approach is that there must be a mechanism to prevent loops on the rings and this is the job of MRP, layer 2 loop prevention.

It is very easy to focus on the ring topology rather than the underlying layer 2 topology described by multiple rings. Design decisions are driven by the same factors as a standard spanning tree network replacing root bridges with ring masters. Traffic patterns at layer 2 are determined by which ring interfaces are forwarding and which are blocking and this in turn should drive design decisions for ring master

placement as well as the direction of RHP flow from the ring masters. Traffic patterns in standard operation as well as failure mode can be determined prior to implementation allowing for appropriate capacity management on all links.

FIGURE 119 Multiple rings with one vlan spanning them



Ring interface ownership

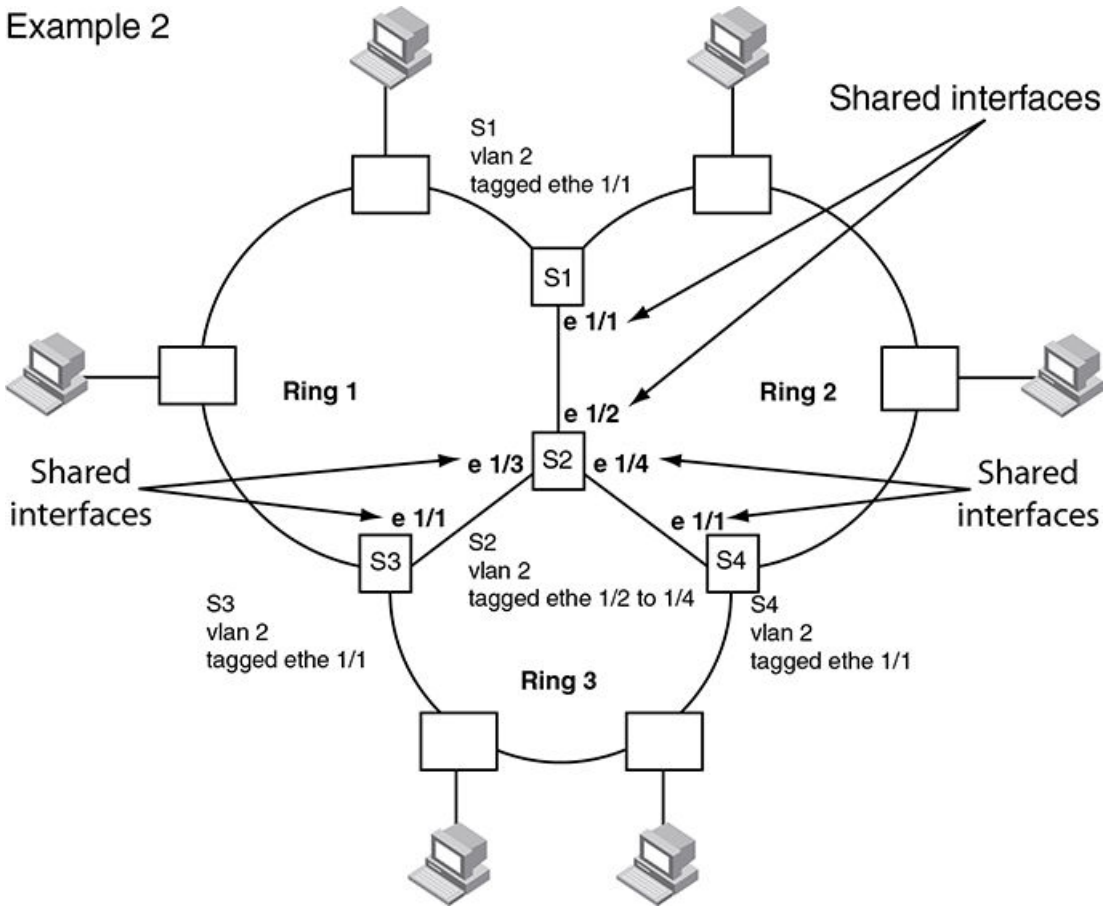
On a shared interface the highest priority ring will be the owner of the interface. In Figure 122 interface e 1/1 on S1 will be owned by ring 1 and marked as a regular interface while in ring 2 the same interface is marked as a tunnel interface in the output of the 'show metro' command.

On S2 interface e 1/2 is again owned by ring 1 and marked as a regular interface.

In Figure 122 the same principles of interface ownership apply. All shared interfaces on ring 1 nodes are shown as owned by ring 1 and marked as regular interfaces. Ring 2 will show shared interfaces as tunnel interfaces.

On S2 e 1/4 and S4 e 1/1 the interfaces will be owned by ring 2, as the highest priority ring on the interface, and ring 3 will show these interfaces as tunnel interfaces.

FIGURE 120 Example 2 multiple rings sharing interfaces - phase 2



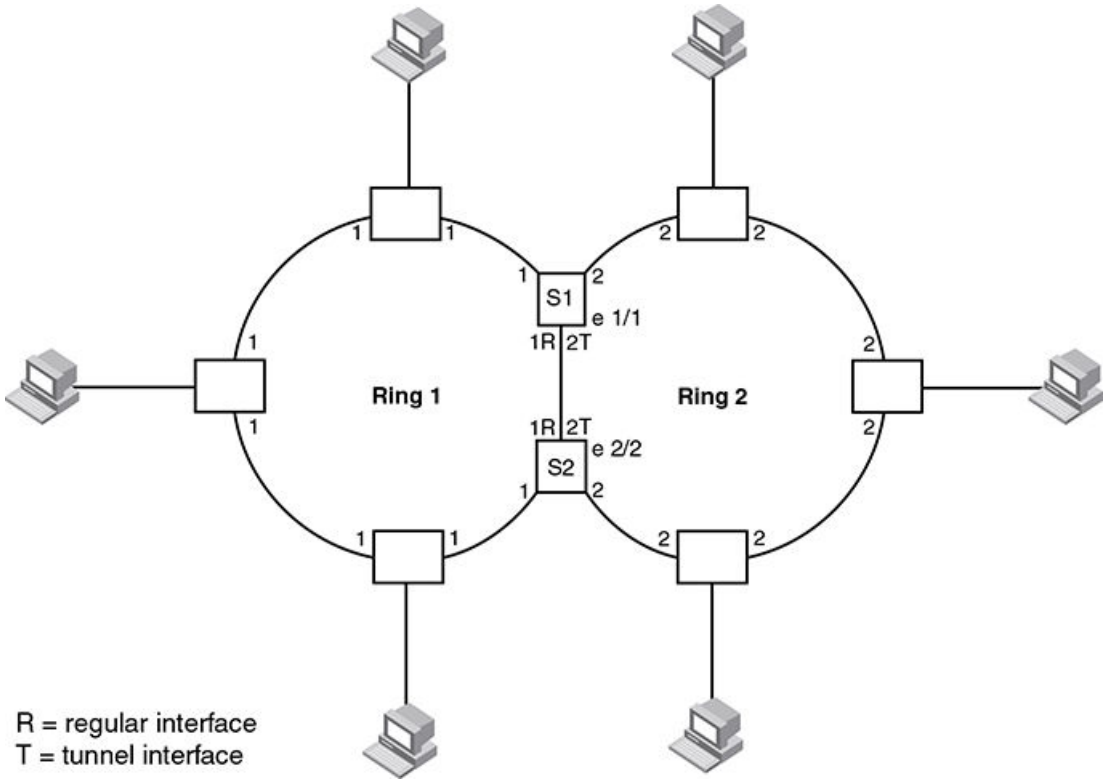
Ring interface IDs and types

For example, in Figure 123, all interfaces configured for ring 1 have a priority of 1. Interface e 1/1 on S1 and e 2/2 on S2 have a priority of 1 since 1 is the highest priority ring that shares the interface.

All interfaces on ring 2, except for e 1/1 on S1 and e 2/2 on S2 have a priority of 2.

If a node has shared interfaces then the ring interfaces that belong to the ring with the highest priority are regular interfaces for that ring and all lower priority ring interfaces are marked as tunnel interfaces. The highest priority ring configured becomes the priority for the interface.

FIGURE 121 Interface IDs and types



In Figure 123, nodes S1 and S2 have interfaces that belong to rings 1 and 2. Interface e 1/1 on S1 and e 2/2 on S2 are regular interfaces for ring 1, but they are tunnel interfaces for ring 2.

Selection of the master node for a ring

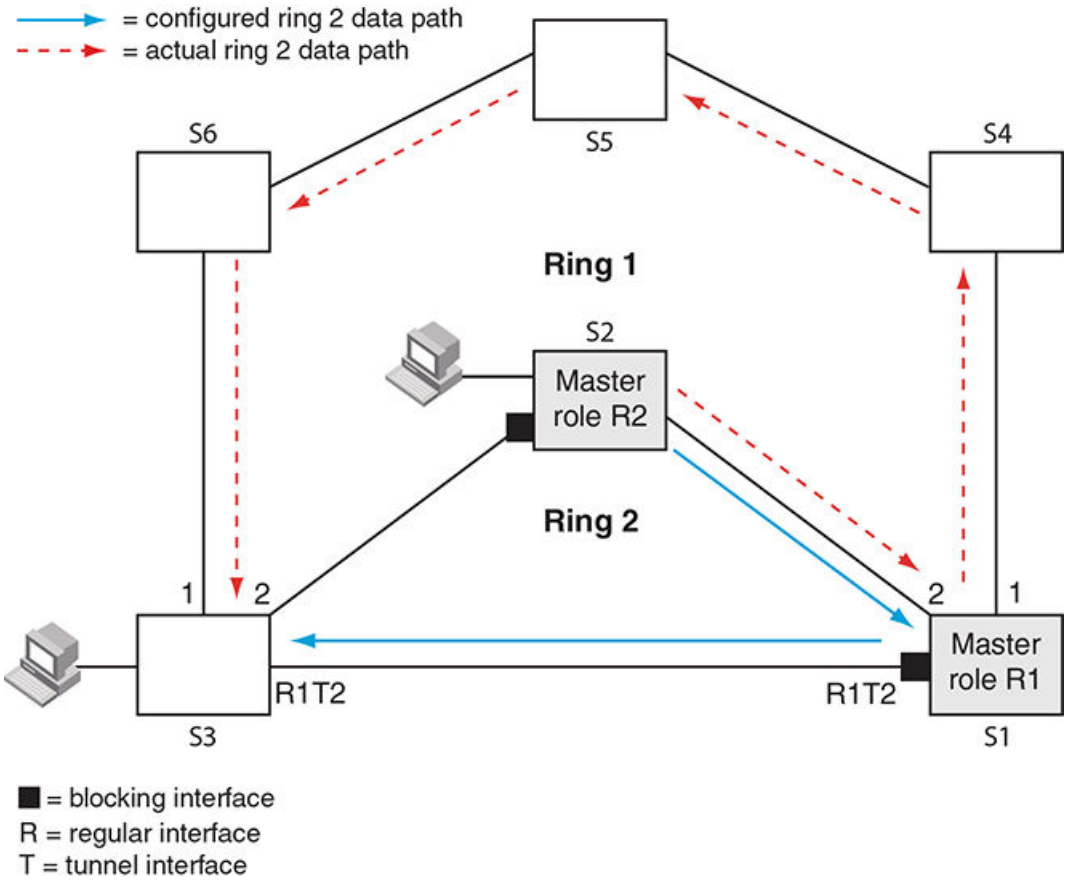
Configuring MRP rings with shared interfaces limits the nodes that can be designated as the master node for any particular ring.

- Any node on the ring that does not have any shared interfaces can be designated as the ring's master.
- You can only designate a node that has shared interfaces as master for a ring where all interfaces for the ring are marked as regular interfaces.
- On a node with shared interfaces, where you configure the role as master, the secondary ring interface should not be a shared interface. If you designate a shared interface as secondary it will be blocking under normal operation and allow RHP's but no data for lower priority rings. This can create unexpected traffic flows on the rings.

In Ring interface IDs and types on page 470 any of the nodes on ring 1, even S1 or S2, can be a master node as all of the ring interfaces, even the shared interfaces between S1 and S2, are marked as regular interfaces for ring 1.

However for ring 2, neither S1 nor S2 can be a master node since the shared interfaces between S1 and S2 are marked as tunnel interfaces for ring 2.

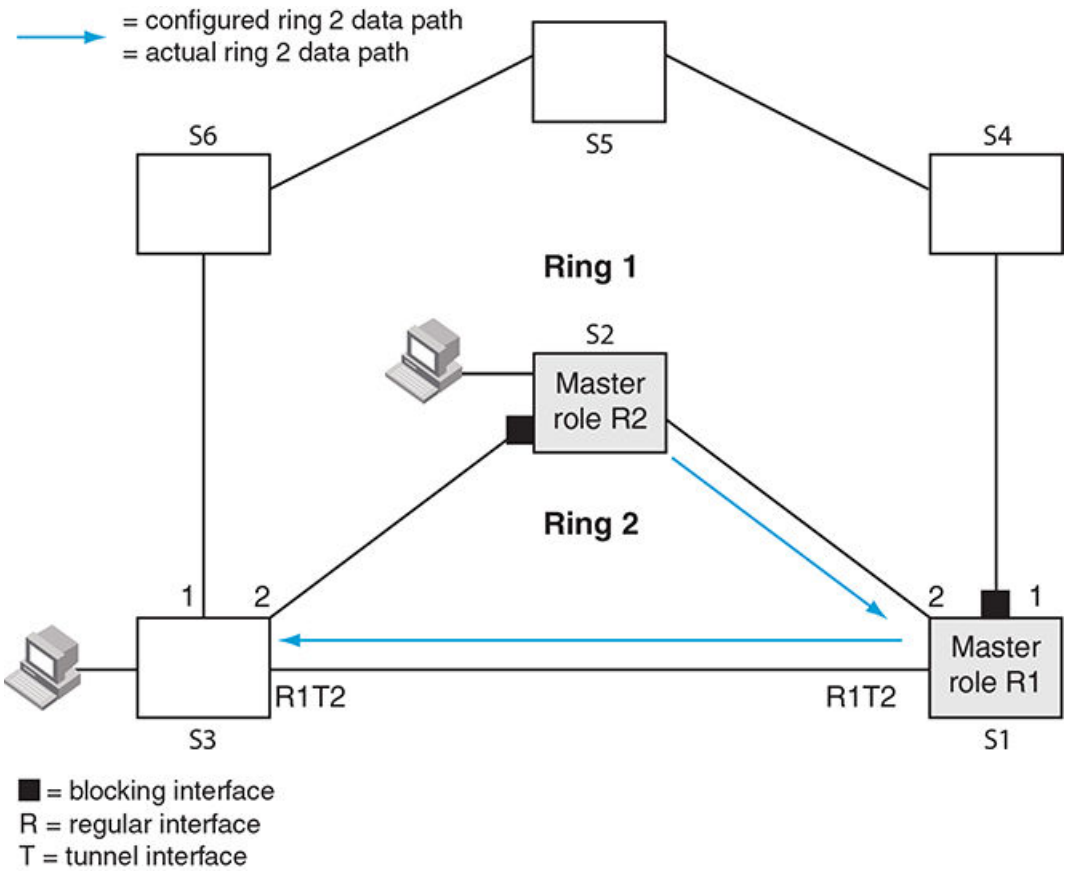
FIGURE 122 Unexpected switching path with shared interface



In Figure 124 ring 2 was configured with shared ring interfaces on S1 and S3 as depicted. S1 was configured as the master for ring 1 and the shared interface was defined as the secondary interface and subsequently blocks data. The designer intended the switching path between a host on S2 and another host on S3 to be via S1 shared interface, however due to the shared interface being blocked the actual switching path becomes S1 to S4,S5,S6 and finally S3.

Ring 2 is still operational but is not behaving in the manner which the design called for. By configuring the secondary interface on the regular port for ring 1 we obtain the expected result as shown in Figure 125.

FIGURE 123 Expected switching path with shared interface



RHP processing in rings with shared interfaces

Interfaces on an MRP ring have one of the following states:

- **Blocking (B)** - The interface can process RHPs, but cannot forward data for the ring. Only the secondary interface on the Master node can be blocking. If the interface receives RHP's for lower priority rings these RHPs will be discarded by this interface. This prevents RHP's from lower priority rings from looping in the topology.
- **Preforwarding (PF)** - The interface will forward RHPs but won't forward data for the ring. All ring interfaces start in this state when you enable MRP. A blocking interface transitions to preforwarding when the preforwarding timer expires.
- **Forwarding (F)** - The interface will forward RHP's and data for the ring. On member switches an interface transitions from preforwarding to forwarding when the preforwarding time expires or the interface receives an RHP with the forwarding bit set. A break in the ring is indicated if the secondary interface on the master fails to receive an RHP within the preforwarding timer and the interface transitions from blocking to forwarding to heal the ring. The preforwarding time is the number of milliseconds the interface will remain in the preforwarding state before changing to the Forwarding state, even without receiving an RHP.

The primary interface of the master node initiates RHP packets and sends them onto the ring. When the packet reaches a forwarding interface, MRP checks to see if the receiving interface is a regular interface or a tunnel interface:

- If the interface is a regular interface, the RHP packet is forwarded to the next interface. Forwarding of the packet continues on the ring until the secondary interface of the master node receives the packet and processes it. For the configured ring the receipt of an RHP with the same ring ID indicates the ring is healthy. RHPs for lower priority rings will be discarded without further processing at this point.

- If the interface is a tunnel interface, MRP checks the priority of the RHP packet and compares it to the priority of the tunnel interface:
  - If the RHP packet's priority is less than or equal to the interface's priority, the packet is forwarded through ring interfaces with higher priority which are in the forwarding state.
  - If the priority of the RHP packet is greater than the priority of the interface, the RHP packet is dropped. For example, if an RHP with a ring ID of 1 arrives at a tunnel interface owned by ring 2 the RHP will be dropped. If an RHP with a ring ID of 2 or 3 arrives at a tunnel interface owned by ring 2 the RHP will be forwarded.

  NOTE
  It is important to understand the key concept of RHPs leaking from lower priority rings to higher priority rings. Always remember that tunnel interfaces check the ring ID of an RHP before forwarding. Higher priority ring ID RHPs will be dropped.

How ring breaks are detected and healed between shared interfaces

If the link between shared interfaces breaks, the secondary interface on the highest priority ring master node changes to a preforwarding state, refer to Flow when a link breaks on page 476. Any RHP from lower priority rings can traverse this interface and thus maintain the integrity of the lower priority rings. When the secondary interface changes state to forwarding the lower priority ring RHP's continue to traverse the interface.

This behavior allows the ring 2 RHP's to continue around ring 1 and back to ring 2 until it reaches the secondary interface on ring 2's master node which changes to blocking mode since it receives its own RHP.

NOTE
On the ring member node, the primary and secondary interface is decided by the RHP flow from the ring master. The secondary interface is always the RHP receiver for its ring RHP's, the primary interface is always the sender of its rings RHP's. If there is no active ring master in the topology, then the running configuration on the member node will show exactly what was configured. This may change on introduction of an active ring master.

Normal flow

Figure 126 and Figure 127 show how RHP packets are forwarded in rings with shared interfaces. Figure 126 shows the flow of ring 1 RHPs while Figure 127 shows how ring 2 RHPs flow.

Interface e 2/1 is the primary interface of the ring 1 master node. The primary interface forwards an RHP packet on the ring. Since all the interfaces on ring 1 are regular interfaces, the RHP packet is forwarded until it reaches interface e 2/2, the secondary interface of the ring 1 master. Receipt of this RHP indicates a healthy ring 1 and interface e2/2 then changes to or maintains its state of blocking.

No copies of the ring 1 RHPs are forwarded on ring 2 tunnel interfaces or ring 2 regular interfaces in accordance with the rule that a higher priority RHP is not permitted to traverse a lower priority ring interface.

FIGURE 124 RHP flow on rings with shared interfaces showing ring 1 RHP flow

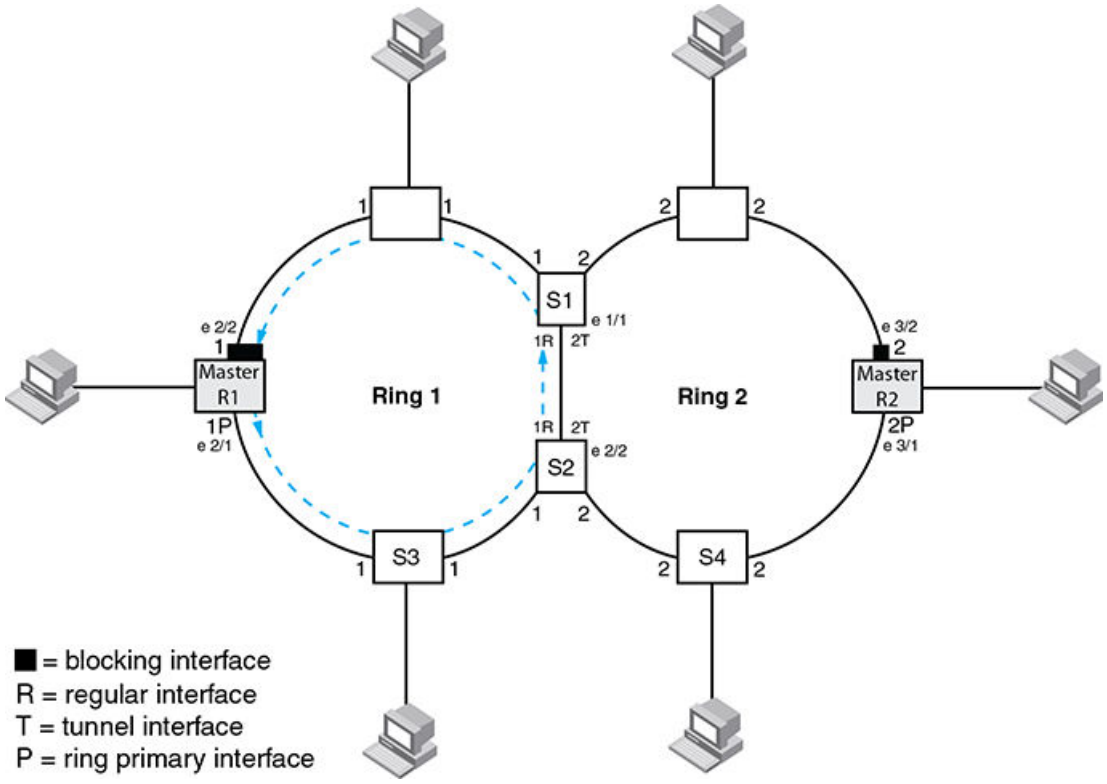
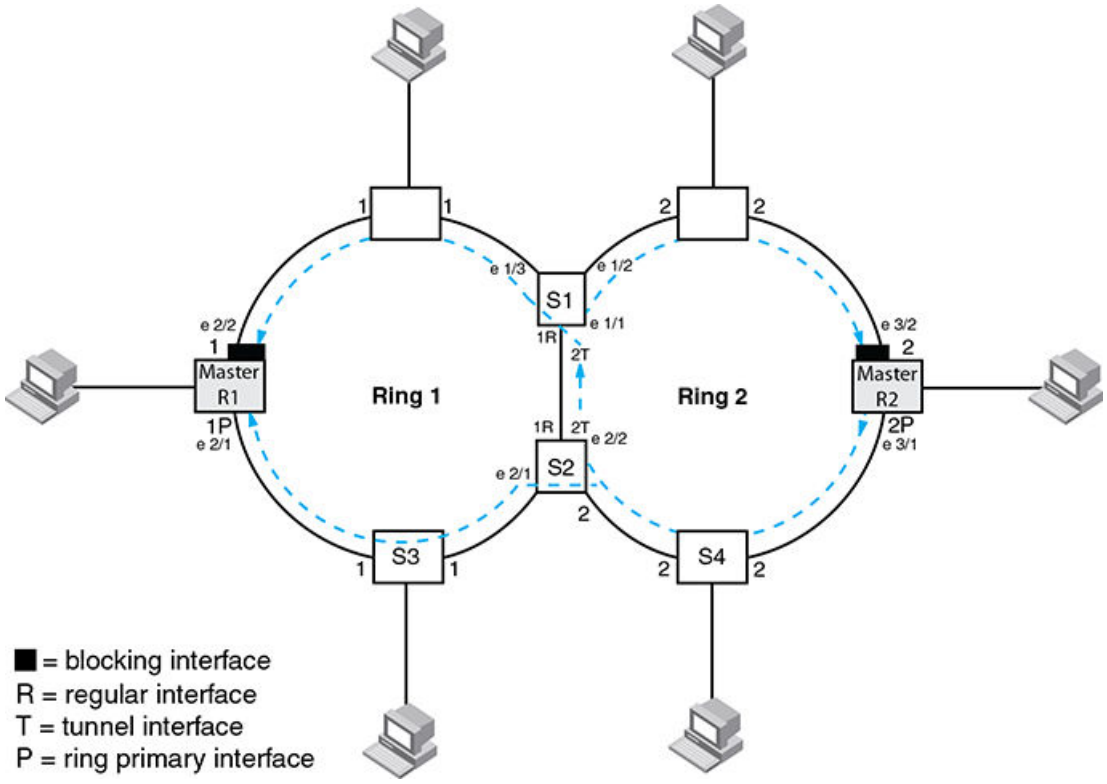


FIGURE 125 RHP flow on rings with shared interfaces showing ring 2 RHP flow



Referring to Figure 127 interface 3/1, is the primary interface of the ring 2 master node. It sends an RHP packet on the ring. Since all interfaces on S4 are regular interfaces, the RHP packet is forwarded on those interfaces.

When the RHP reaches S2:

- A copy of the RHP is sent out of regular interface e 2/1 onto ring 1. This is in accordance with the rule that a lower priority RHP can traverse a higher priority ring interface. This RHP is forwarded until it reaches the ring 1 master where it is discarded.
- A copy of the RHP is forwarded out of the ring 2 tunnel interface on e 2/2

The RHP is received by S1 on e 1/1 and then:

- A copy of the RHP is sent out of regular interface e 1/2 on ring 2
- A copy of the RHP is sent out of regulate interface e 1/3 on ring 1. This is in accordance with the rule that a lower priority RHP can traverse a higher priority ring interface. This RHP is forwarded until it reaches the ring 1 master where it is discarded.

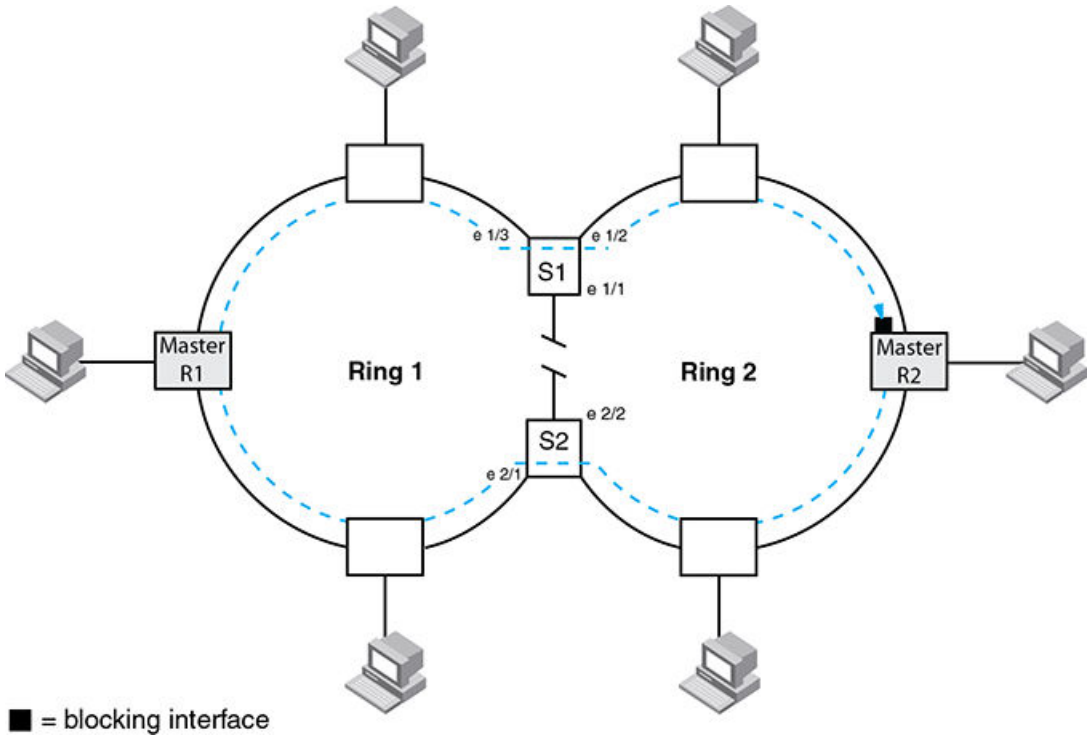
Flow when a link breaks

Referring to Figure 128 if the link between S1 and S2 fails, the secondary interface on the ring 1 master node changes to a forwarding state.

The RHPs from the master for ring 2 reach S2 and a copy of the RHP is forwarded out of e 2/1. This RHP traverses the ring 1 master and continues around ring 1 until it reaches S1. After S1 the RHP is back on ring 2 and is finally received by the master for ring 2 which keeps its secondary interface in blocking mode.

It should now be clear how the flow of lower priority RHPs over the higher priority ring ensure that both ring masters do not transition to forwarding and create a loop condition.

FIGURE 126 Flow of RHP packets when a link for shared interfaces breaks



Ring 2 RHPs follow this path until the link is restored. Once the link is restored the ring 1 master will transition its secondary ring interface to blocking and the ring 2 RHP flow is as shown in Normal flow on page 474.

NOTE
There should always be a layer 2 protocol configured in the default vlan when MRP is configured with all dual mode ports.

Configuring MRP with shared interfaces

MRP Phase 2 allows you to enter commands such as the following when configuring MRP.

```
device(config)# vlan 2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name CustomerA
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2-mrp-1)# metro-ring 2
device(config-vlan-2-mrp-2)# name CustomerB
device(config-vlan-2-mrp-2)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
```

Syntax: [no] metro-ring ring-id

The *ring-id* parameter specifies the ring ID, which can be from 1 - 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] name string

The *string* parameter specifies a name for the ring. The name is optional, but it can have up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] ring-interface ethernet

The **ethernet** *primary-if* parameter specifies the primary interface. On the master node, the primary interface is the one that originates RHPs. Ring control traffic and layer 2 data traffic will flow in the outward direction from this interface by default. On member nodes, the direction of traffic flow depends on the traffic direction selected by the master node. Therefore, on a member node, the order in which you enter the interfaces does not matter.

The **ethernet***secondary-if* parameter specifies the secondary interface.

Syntax: [no] enable

The **enable** command enables the ring.

Tuning MRP timers

To effectively tune MRP timers it is crucial to understand the association between the hello time and the preforwarding time.

Flushing the mac table following an MRP event

After an MRP event switches in the ring flush MAC tables and relearn to ensure correct forwarding paths. Notification to flush is carried out by sending topology change RHP's.

Hello time

This timer specifies the interval at which RHP's are generated by the ring master. It should be noted that this interval is applied not only to standard RHP's but also to topology change notification RHP's. For example: Setting the hello time to its maximum value of 15,000 ms would mean that the three topology change notification RHP's that are sent following a ring break being detected or a ring heal event would result in MAC table flushes three times at 15 second intervals. On a busy network this would cause unnecessary impact.

Preforwarding time

The preforwarding time defines the amount of time an interface will take to move from blocking to preforwarding without RHP's being received. It also defines the amount of time an interface will take to move from preforwarding to forwarding without RHP's being received.

The preforwarding time must be at least 2 x hello time and must be a multiple of the hello time.

The preforwarding time for a lower priority ring must be greater than or equal to the highest higher priority ring.

For example: Setting the preforwarding time to its maximum value of 30,000 ms will mean that a break in the ring (assuming no alarm RHP's are generated) will take one minute to heal.

Setting hello and preforwarding timers appropriately

When setting timers both the hello time and the preforwarding time should be considered to ensure that the appropriate recovery time is applied on the network.

Consider a break in the network that does not generate alarm RHP's

Example 1(default values):

Preforwarding time = 300ms

Hello time = 100ms

Time to forwarding = 2 x preforwarding time = 600ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 900ms = 0.9secs

Example 2:

Preforwarding time = 10000ms

Hello time = 100ms

Time to forwarding = 2 x preforwarding time = 20000ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 20300ms = 20.3secs

Example 3:

Preforwarding time = 10000ms

Hello time = 5000ms

Time to forwarding = 2 x preforwarding time = 20000ms

Post recovery mac table flush time = 3 x hello time = 15000ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 35000ms = 35secs

It can therefore be seen that the hello time should not be changed on the network unless there is evidence of regular misses on the ring. Time to traverse the ring can be determined by running MRP diagnostics.

Effect of the scale timer

Changing the scale timer has a significant effect on the operation of MRP and should be considered for very high performance low latency networks where a very rapid failure detection and recovery mode is required. Achieving this rapid detection and recovery requires very stable high speed environments to prevent a high level of unnecessary topology changes in the environment.

The effect of setting the scale timer is that the time taken to move from blocking to preforwarding and preforwarding to forwarding is (preforwarding value - the hello time). This is a significant change to the operation of MRP in the default state which has been described in the previous section.

Note: When setting the timer at the CLI the actual value used will be exactly half of the input value. The examples that follow assume the corrected value.

Consider a break in the network that does not generate alarm RHP's

Example 1(default values):

Preforwarding time = 300ms

Hello time = 100ms

Time to forwarding = preforwarding time - hello time = 200ms

Post recovery mac table flush time = 3 x hello time = 300ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 500ms = 0.5secs

Example 2:

Preforwarding time = 100ms

Hello time = 50ms

Time to forwarding = preforwarding time - hello time = 50ms

Post recovery mac table flush time = 3 x hello time = 150ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 200ms = 0.2secs

Example 3:

Preforwarding time = 10000ms

Hello time = 5000ms

Time to forwarding = preforwarding time - hello time = 5000ms

Post recovery mac table flush time = 3 x hello time = 15000ms

Full connectivity = Time to forwarding + Post recovery mac table flush time = 20000ms = 20sec

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.

```
device(config-vlan-2-mrp-1)#diagnostics
```

Syntax: [no] diagnostics

NOTE
When using the 'show metro' command, the member node of a ring does not display correctly since the MRP RHPs are hardware forwarded (or software forwarded on the linecard), these statistics are only reflective of the MRP RHPs that made it to the management processor. In most cases, these would be TC RHPs since the MP needs to flush MACs in that case.

NOTE
This command is valid only on the master node.

Displaying MRP diagnostics

To display MRP diagnostics results, enter the following command on the Master node.

```
device(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 62 CLI display of MRP ring diagnostic information

This field...	Displays...
Ring id	The ring ID.
Diag state	The state of ring diagnostics.
RHP average time	The average round-trip time for an RHP packet on the ring. The calculated time has a granularity of 1 microsecond.
Recommended hello time	The hello time recommended by the software based on the RHP average round-trip time.
Recommended Prefwing time	The preforwarding time recommended by the software based on the RHP average round-trip time.
Diag frame sent	The number of diagnostic RHPs sent for the test.
Diag frame lost	The number of diagnostic RHPs lost during the test.

If the recommended hello time and preforwarding time are different from the actual settings and you want to change them, refer to Configuring MRP on page 464.

Displaying MRP information

You can display the following MRP information:

- Topology group ID associated with the MRP ring
- Ring configuration information and statistics

Displaying topology group information

To display topology group information, enter the following command.

Syntax: show topology-group [**group-id**]

Displaying ring information

To display ring information, enter the following command.

```
device(config)# show metro
Metro Ring 10 - VLAN Type REGULAR
==============
Ring       State      Ring       Master     Topo       Hello      Prefwing
id                    role       vlan       group      time(ms)   time(ms)
10         enabled    member     7          1          100        300
Ring interfaces Interface role  Interface state interface type
ethernet 1/1    primary         forwarding      regular
ethernet 30/1   secondary       forwarding      regular
RHPs sent       RHPs rcvd       TC rcvd    TC sent    State changes
0               0               69         0          6
```

Syntax: show metro [ring-id]

This display shows the following information.

TABLE 63 CLI display of MRP ring information

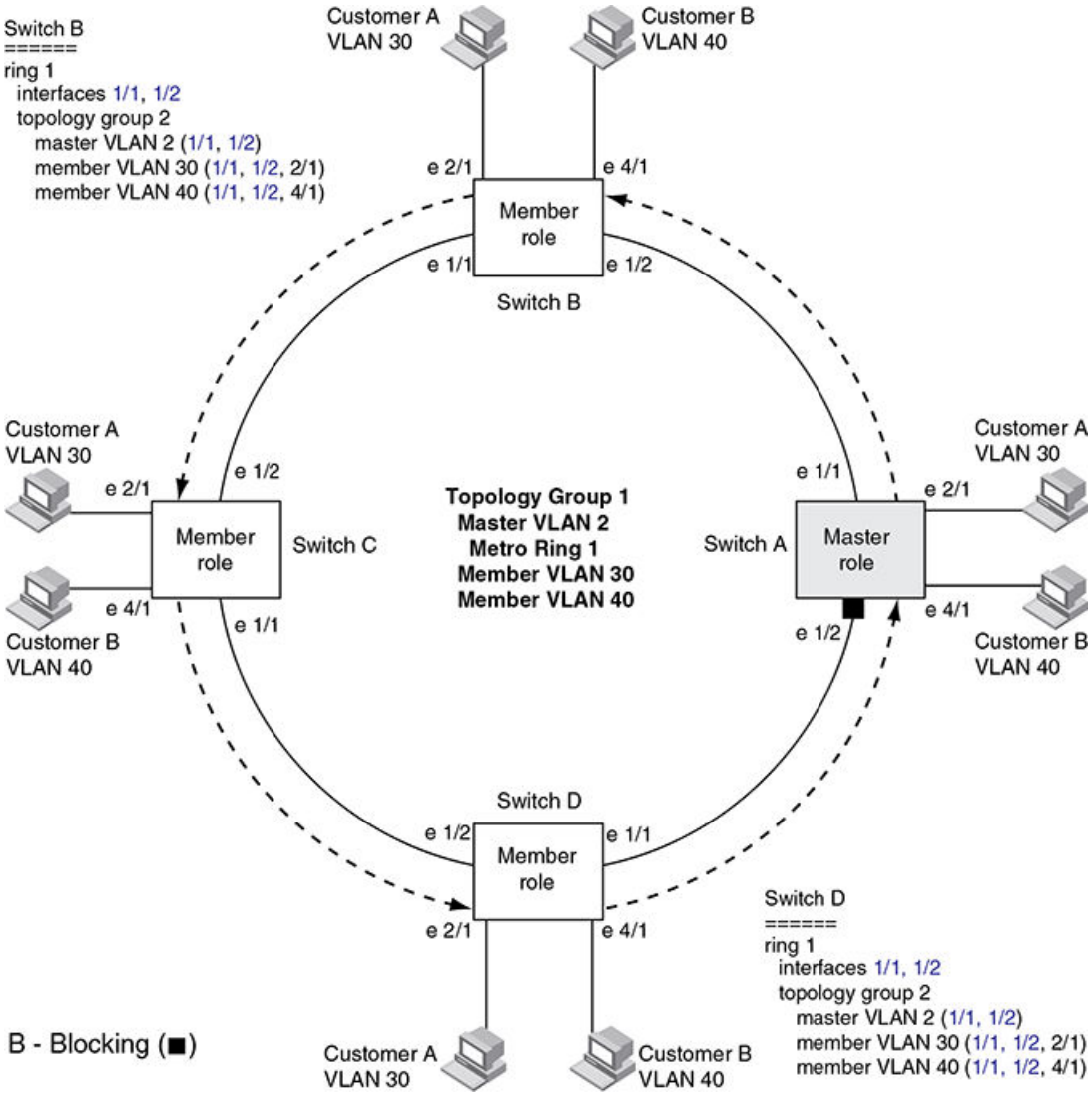
This field...	Displays...
Ring id	The ring ID
State	The state of MRP. The state can be one of the following: • enabled - MRP is enabled • disabled - MRP is disabled
Ring role	Whether this node is the master for the ring. The role can be one of the following: • master • member
Topo group	The topology group ID if a topology group is configured. This field will show the value 'not conf' if no topology group is in use.
Hello time	The interval, in milliseconds, which the forwarding port on the ring's master node sends Ring Hello Packets (RHPs). Configured in increments of 100ms.
Prefwing time	The number of milliseconds a MRP interface will wait to move an interface in blocking state to preforwarding state if no RHP's are received. It is also the number of milliseconds that an interface that has entered the preforwarding state will wait before changing to the forwarding state. If a member port in the preforwarding state does not receive an RHP within the preforwarding time (Prefwing time), the port assumes that a topology change has occurred and changes to the forwarding state. The secondary port on the master node changes to blocking if it receives an RHP, but changes to forwarding if the port does not receive an RHP before the preforwarding time expires. Configured in increments of 100ms. **NOTE** A member node's preforwarding interface also changes from preforwarding to forwarding if it receives an RHP whose forwarding bit is on.
Ring interfaces	The device's two interfaces with the ring. **NOTE** If the interfaces are trunk groups, only the primary ports of the groups are listed.
Interface role	The interface role can be one of the following: • primary – Master node - The interface generates RHPs. – Member node - The interface forwards RHPs received on the other interface (the secondary interface). • secondary - The interface does not generate RHPs. – Master node - The interface listens for RHPs. – Member node - The interface receives RHPs.
Forwarding state	Whether MRP Forwarding is enabled on the interface. The forwarding state can be one of the following: • blocking - The interface is blocking layer 2 data traffic and RHPs

TABLE 63 CLI display of MRP ring information (continued)

This field...	Displays...
	• disabled - The interface is down • forwarding - The interface is forwarding layer 2 data traffic and RHPs • preforwarding - The interface is listening for RHPs but is blocking layer 2 data traffic
Interface Type	Shows if the interface is a regular port or a tunnel port.
RHPs sent	The number of RHPs sent on the interface.
RHPs rcvd	The number of RHPs received on the interface.
TC RHPs rcvd	The number of Topology Change RHPs received on the interface.
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 129.



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.

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# master
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2-mrp-1)# exit
device(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.

```
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(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.

```
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
device(config-topo-group-1)# member-vlan 30
device(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.

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2)# exit
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Commands on Switch C

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2)# exit
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
```

```
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Commands on Switch D

```
device(config)# vlan 2
device(config-vlan-2)# tag ethernet 1/1 to 1/2
device(config-vlan-2)# metro-ring 1
device(config-vlan-2-mrp-1)# name "Metro A"
device(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-vlan-2-mrp-1)# enable
device(config-vlan-2)# exit
device(config)# vlan 30
device(config-vlan-30)# tag ethernet 1/1 to 1/2
device(config-vlan-30)# tag ethernet 2/1
device(config-vlan-30)# exit
device(config)# vlan 40
device(config-vlan-40)# tag ethernet 1/1 to 1/2
device(config-vlan-40)# tag ethernet 4/1
device(config-vlan-40)# exit
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 2
device(config-topo-group-1)# member-vlan 30
device(config-topo-group-1)# member-vlan 40
```

Configuring MRP under an ESI VLAN

MRP can also be configured under a vlan that is part of a user-configured ESI. Configuring MRP in this scenario is exactly the same as explained before.

```
device(config)# esi customer1 encapsulation cvlan
device(config-esi-customer1)# vlan 100
device(config-esi-customer1-vlan-100)# tag ethernet 1/1 to 1/2
device(config-esi-customer1-vlan-100)# metro-ring 1
device(config-esi-customer1-vlan-100-mrp-1)# name "Metro A"
device(config-esi-customer1-vlan-100-mrp-1)# master
device(config-esi-customer1-vlan-100-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
device(config-esi-customer1-vlan-100-mrp-1)# enable
device(config-esi-customer1-vlan-100-mrp-1)# exit
device(config-esi-customer1-vlan-100)# exit
```

Configuration considerations

The configuration considerations are as follows:

- MRP can be configured for vlans with encapsulation type B-VLAN, S-VLAN or C-VLAN.
- When MRP is configured for vlans under an ESI, the MRP members must be part of the same ESI.

Ethernet Ring Protection Protocol

Ethernet Ring Protection

Ethernet Ring Protection (ERP), a non-proprietary protocol described in ITU-T G.8032 (Version 1 and 2), integrates an Automatic Protection Switching (APS) protocol and protection switching mechanisms to provide Layer 2 loop avoidance and fast reconvergence in Layer 2 ring topologies. ERP supports multi-ring and ladder topologies. ERP can also function with IEEE 802.1ag to support link monitoring when non-participating devices exist within the Ethernet ring.

You can enable one instance of ERP on a device. Changes to a master VLAN apply to the member VLANs.

> NOTE
> Before configuring ERP, you must configure a VLAN and the ports you require for your deployment.

This chapter describes ERP components, features, and how to configure, and manage ERP.

Ethernet Ring Protection components

An ERP deployment consists of the following components:

- Roles assigned to devices, called Ethernet Ring Nodes (ERN)
- Interfaces
- Protocols -- ERP alone or with IEEE 802.1ag
- ERP messaging
- ERP operational states
- ERP timers

ERN roles

In an Ethernet ring topology you can assign each ERN one of three roles:

- **Ring Protection Link Owner (RPL owner)** -- One RPL owner must exist in each ring; its role is to prevent loops by maintaining a break in traffic flow to one configured link while no failure condition exists within the ring.

- **Non-RPL node** -- Multiple non-RPL nodes, can exist in a ring; but they have no special role and perform only as ring members. Ring members apply and then forward the information received in R-APS messages.
- **Ring Protection Link (RPL) node** -- RPL nodes block traffic to the segment that connects to the blocking port of the RPL owner. The RPL node is used in dual-end blocking and is part of the FDB optimization feature.

Each device can only have one role at any time. Non-ERN devices can also exist in topologies that use IEEE 802.1ag.

ERN interfaces

In addition to a role, each ERN has two configured interfaces:

- Left interface
- Right interface

Traffic enters one interface (ingress) and exits the device using the other interface (egress). The right and left interfaces are physically connected.

You must configure these left and right interfaces in the same pattern across all ERNs within a topology. For example you can assign the interfaces as left/right, left/right, left/right, and so on. It is not acceptable, however, to assign interfaces in random order, such as left/right in the configuration of one ERN and then right/left in the configuration of the next ERN.

Protocols

You can configure standalone ERP or ERP with IEEE 802.1ag support.

Using standalone ERP

When using standalone ERP, all devices have a role, and all devices participate at least as ERP members.

Ring-APS (R-APS) messages are sent at initial start-up of a configuration and periodically when link or node failures or recoveries occur. Each ERN applies the information received in the R-APS messages and forwards the received RAPS messages if both ports are in the forwarding state.

The sending ERN terminates the message when it receives a message originally sent from itself.

Configurable timers prevent ERNs from receiving outdated messages and decrease failure reporting time to allow increased stability within the topology.

To properly configure and troubleshoot ERP, an understanding of the messaging, operational states, and timers is essential. For more information about the ERP protocol, see ITU-T G.8032.

Using ERP with IEEE 802.1ag support

When you have other nonparticipating switches in the ring, you can use the IEEE 802.1ag support to perform link health checks to the next ERN.

With IEEE 802.1ag configured, the ERNs within the ring send Continuity Check Messages (CCM) to verify the integrity of their own links. If a node is not receiving CCMs or if a link goes down, a failure is reported to the ring through R-APS messages. See Figure 130.

FIGURE 127 ERP with IEEE 802.1ag support

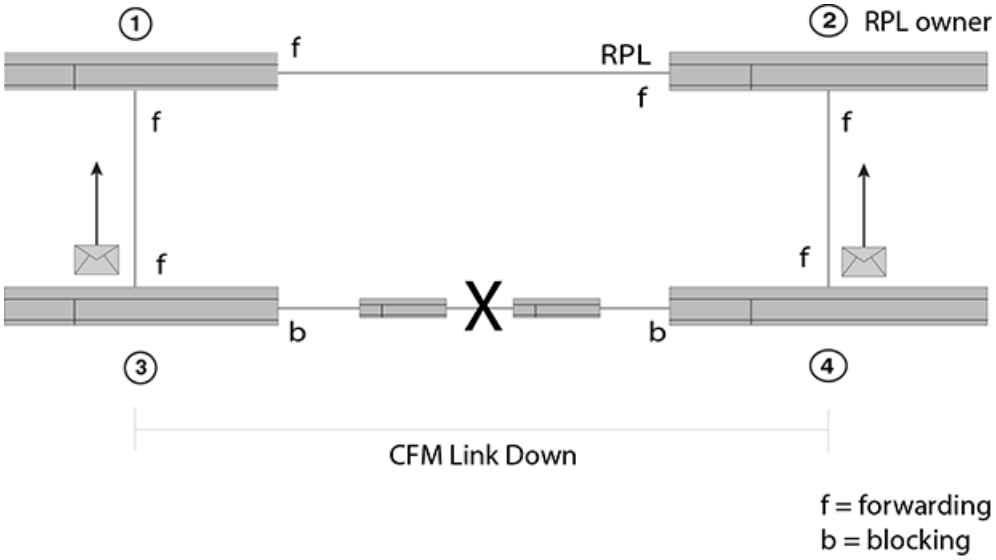


Figure 130 shows a segment with ERNs 3 and 4 and two non-participating switches located on the same network segment between them. When ERNs 3 and 4 stopped receiving CCMs, the following actions occurred on ERNs 3 and 4:

1. Blocked the failed port
2. Transmitted a R-APS (SF) message
3. Unblocked the non-failed port
4. Flushed the FDB
5. Entered the Protection state

   As a result, ERN 2, the RPL owner, unblocked the RPL, and the topology became stable and loop free.

ERP messaging

In ERP, ERNs send R-APS messages. The figure below shows the general R-APS packet structure. For details about the packet structures, see ITU-T G.8032.

Figure to be added here for R-APS packet structure.

The destination MAC address (Dst Mac) is the first element in the packet and is of the form 00-00-00-00-00-<ERP ID>. The default value is 01. However, you can configure the ERP ID with the **raps-default-mac** command. In ITU-T G.8032 Version 1 the default value is always used.

The Node ID indicates the base MAC address and can be found in the R-APS specific information part of a R-APS message.

ERP operational states

RPL nodes can be in one of six different states in Version 2:

- Init
- Idle
- Protection state, which is designated as a signal fail (SF) event in the R-APS

- Manual-switch (MS)
- Forced-switch (FS)
- Pending (not available if using Version 1)

When an ERP topology starts up, each ERN (in Init state) transmits a R-APS (NR). After start-up, the behavior varies by assigned role. ERP operational states shows the initialization process for an ERN.

Message exchange and actions during ERN initialization version 2

RPL owner	Non-RPL node	RPL node
Init state	Init state	Init state
1. Blocks the RPL 2. Sends a R-APS (NR) 3. Enters the Pending state.	1. Blocks the left interface 2. Sends a R-APS (NR) 3. Enters the Pending state	1. Blocks the left interface 2. Sends a R-APS (NR) 3. Enters the Pending state
4. Starts the WTR timer 5. (After the WTR expires) stops sending NR 6. Sends R-APS (NR, RB, DNF) 7. Enters the Idle state	After receiving the (NR, RB, DNF) from the RPL owner: 1. Unblocks the non-failed blocking port 2. Stops sending (NR) 3. Enters the Idle state	After receiving the (NR, RB, DNF) from the RPL owner: 1. Blocks the RPL port 2. Unblocks the other ports 3. Enters the Idle state

When the ring is in the Pending state, an ERN flushes the filtering database (FDB) if it receives any of the following state requests:

- Signal-fail (SF)
- No request (NR), RPL Blocked (RB)

NOTE
ITU-T G.8032 Version 1 does not use a Pending state, so from the Protection state ERNs enter the Idle state.

ERP timers

ERP provides various timers to ensure stability in the ring while a recovery is in progress or to prevent frequent triggering of the protection switching. All of the timers are operator configurable.

- **Guard timer** -- All ERNs use a guard timer. The guard timer prevents the possibility of forming a closed loop and prevents ERNs from applying outdated R-APS messages. The guard timer activates when an ERN receives information about a local switching request, such as after a switch fail (SF), manual switch (MS), or forced switch (FS). When this timer expires, the ERN begins to apply actions from the R-APS it receives. This timer cannot be manually stopped.
- **Wait to restore (WTR) timer** -- The RPL owner uses the WTR timer. The WTR timer applies to the revertive mode to prevent frequent triggering of the protection switching due to port flapping or intermittent signal failure defects. When this timer expires, the RPL owner sends a R-APS (NR, RB) through the ring.
- **Wait to Block (WTB) timers** -- This wait-to-block timer is activated on the RPL owner. The RPL owner uses WTB timers before initiating an RPL block and then reverting to the idle state after operator-initiated commands, such as for FS or MS conditions, are entered. Because multiple FS commands are allowed to co-exist in a ring, the WTB timer ensures that the clearing of a single FS command does not trigger the re-blocking of the RPL. The WTB timer is defined to be 5 seconds longer than the guard timer, which is enough time to allow a reporting ERN to transmit two R-APS messages and allow the ring to identify the latent condition.When clearing a MS command, the WTB timer prevents the formation of a closed loop due to the RPL owner node applying an outdated remote MS request during the recovery process.
- **Hold-off timer** -- Each ERN uses a hold-off timer to delay reporting a port failure. When the timer expires, the ERN checks the port status. If the issue still exists, the failure is reported. If the issue does not exist, nothing is reported.

- **Message interval** -- This is an operator configurable feature for sending out R-APS messages continuously when events happen.

Initializing a new ERN

A newly configured Version 2 ERP topology with four ERNs initializes as described in this section. The ERNs have the following roles:

- ERN 2 is the RPL owner.
- ERNs 1, 3, and 4 are non-RPL nodes.

Figure 131 shows the first step of initialization beginning from ERN 4, a non-RPL node. The actions of each ERN are:

- ERN 1 takes no action. Both ports are in the forwarding state.
- ERN 2 (RPL owner) takes no action. Both ports, including the RPL port, are in the VLAN port forwarding state.
- ERN 3 takes no action. Both ports are in the forwarding state.
- From the Init state ERN 4 stops all timers (guard, WTR, WTB), blocks the left port, unblocks the right port, transmits R-APS (NR) messages, and enters the Pending state.

FIGURE 128 Initializing an ERN topology - I

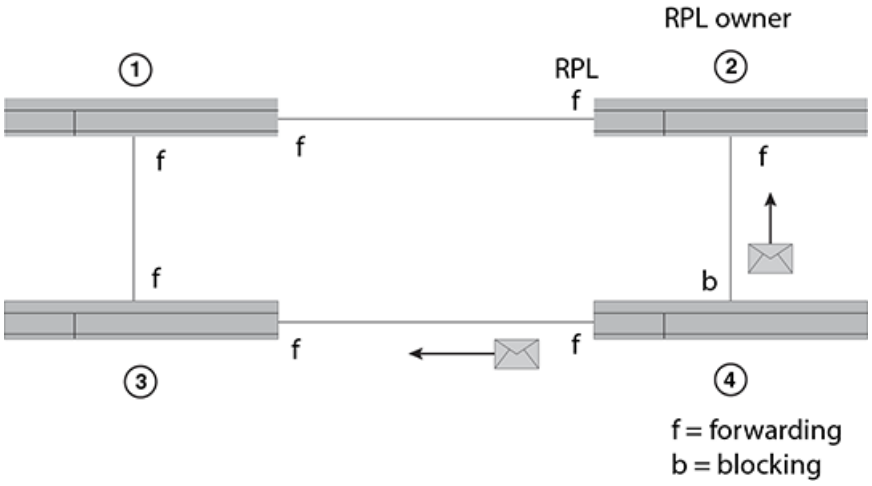


Figure 132 shows the next sequence of events. Next, ERN 1 initializes. The actions of each ERN are:

- ERN 1 stops all timers (guard, WTR, WTB), blocks the left port, unblocks the right port, transmits R-APS (NR) messages, and enters the Pending state.
- ERN 2 takes no action. Both ports are in the forwarding state.
- ERN 3 takes no action. Both ports are in the forwarding state.
- ERN 4 stays in the Pending state, transmits R-APS (NR) messages, and continues to block the left interface.

FIGURE 129 Initializing an ERP topology - II

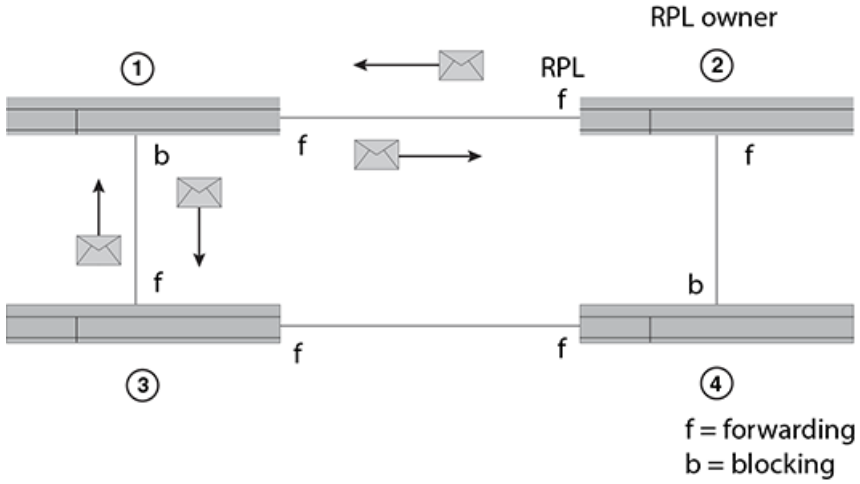


Figure 133 shows the next sequence of events. The actions of each ERN are:

- ERN 1 terminates R-APS received on the blocked port, unblocks the non-failed port, stops transmitting R-APS (NR) messages, and enters the Pending state.
- ERN 2 takes no action.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages.

FIGURE 130 Initializing an ERP topology - III

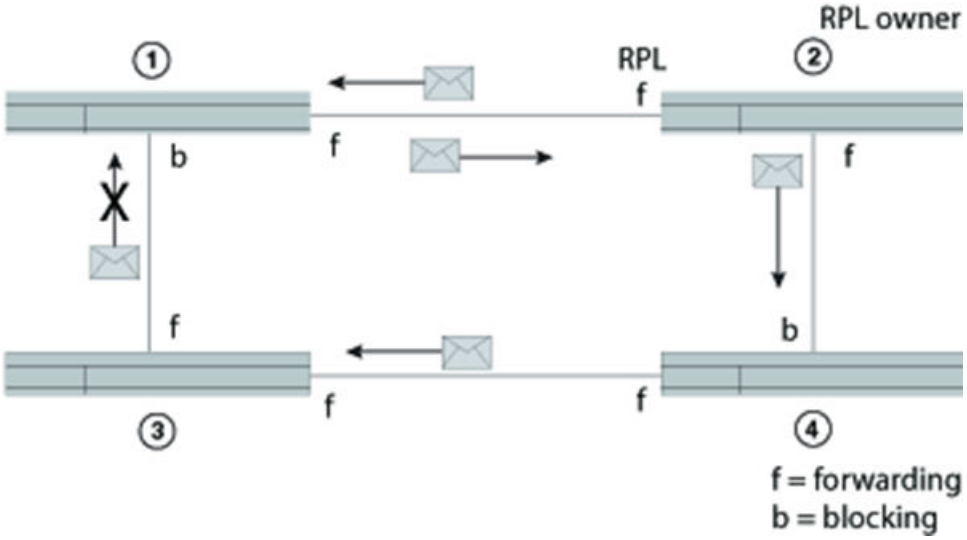


Figure 134 shows the next sequence of events. The actions of each ERN are:

- ERN 1, from the Pending state, unblocks the left interface, stops sending R-APS (NR) and stays in the Pending state. Now both interfaces are in the forwarding state.
- ERN 2 takes no action.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages. The left interface is blocked, and the right interface is in the forwarding state.

FIGURE 131 Initializing an ERP topology - IV

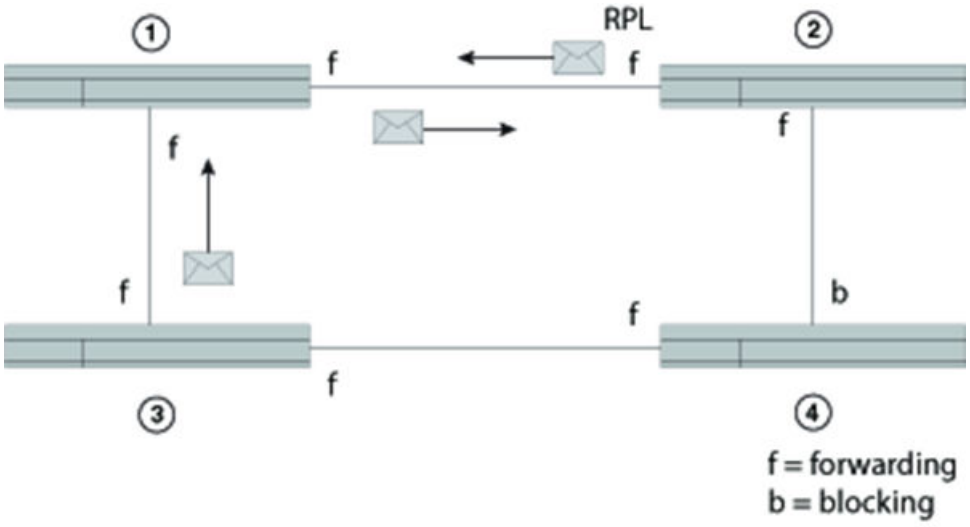


Figure 135 shows the next sequence of events. Next ERN 2 initializes. The actions of each ERN are:

- ERN 1 stays in the Pending state.
- ERN 2 (RPL owner), from the Init state, stops the guard timer, stops the WTB timer, blocks the RPL, unblocks the non-RPL port, enters the Pending state, transmits R-APS (NR) messages, and starts the WTR timer.
- ERN 3 takes no action.
- ERN 4 stays in the Pending state and transmits R-APS (NR) messages. The left interface is blocked.

FIGURE 132 Initializing an ERP topology - V

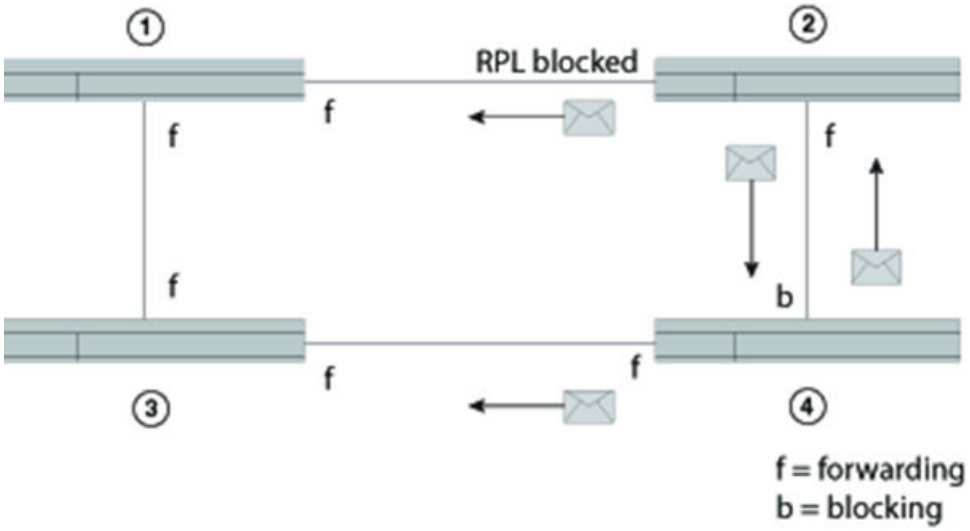
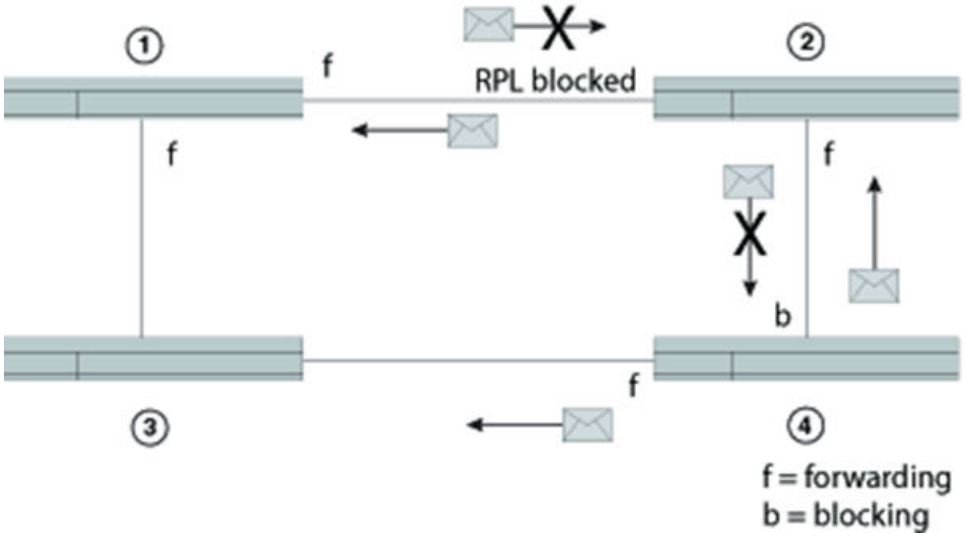


Figure 136 shows the next sequence of events. The actions of each ERN are:

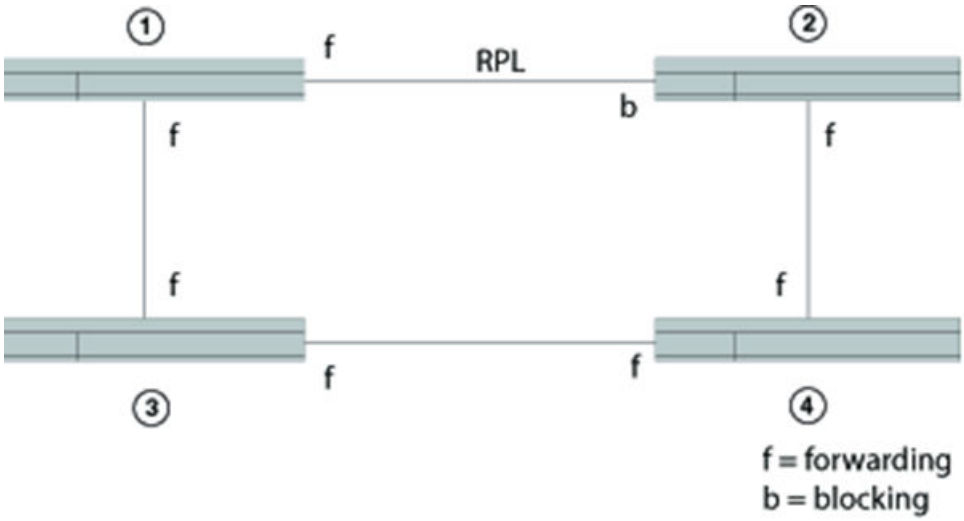
- After the WTB timer expires, ERN 2 (RPL owner in the Pending state) transmits R-APS (NR, RB), and then ERN 2 enters the Idle state.
- ERN 1, still in the Pending state, forwards R-APS (NR, RB) and enters the Idle state.
- ERN 3 takes no action.
- ERN 4 from the Pending state and stops transmitting R-APS (NR).

FIGURE 133 Initializing an ERP topology - VI



Lastly, ERNs 1, 2, and 3 are in the Idle state, and ERN 4 changes the blocking port to the forwarding state. All ERNs remain in the Idle state. See Figure 137.

FIGURE 134 Initializing an ERP topology - VII



Signal fail

Signal fail and signal fail recovery provide the mechanism to repair the ring to preserve connectivity among customer networks.

ERP guarantees that although physically the topology is a ring, logically it is loop-free. One link, called the Ring Protection Link (RPL), is blocked to traffic. When a non-RPL link fails in the ring, the signal failure mechanism triggers and causes the RPL to become forwarding. Later, signal fail recovery can occur to restore the ring to the original setup.

Convergence time is the total time that it takes for the RPL owner to receive the R-APS (NR) message and block the RPL port until the ERN with the failed link receives notice and unblocks the failed link.

Figure 138 shows a simple Ethernet ring topology before a failure. This diagram shows dual-end blocking enabled (thick line) between ERNs one (RPL node) and 6 (RPL owner). ERNs 3, 2, 4, and 5 are non-RPL nodes.

FIGURE 135 ERP topology

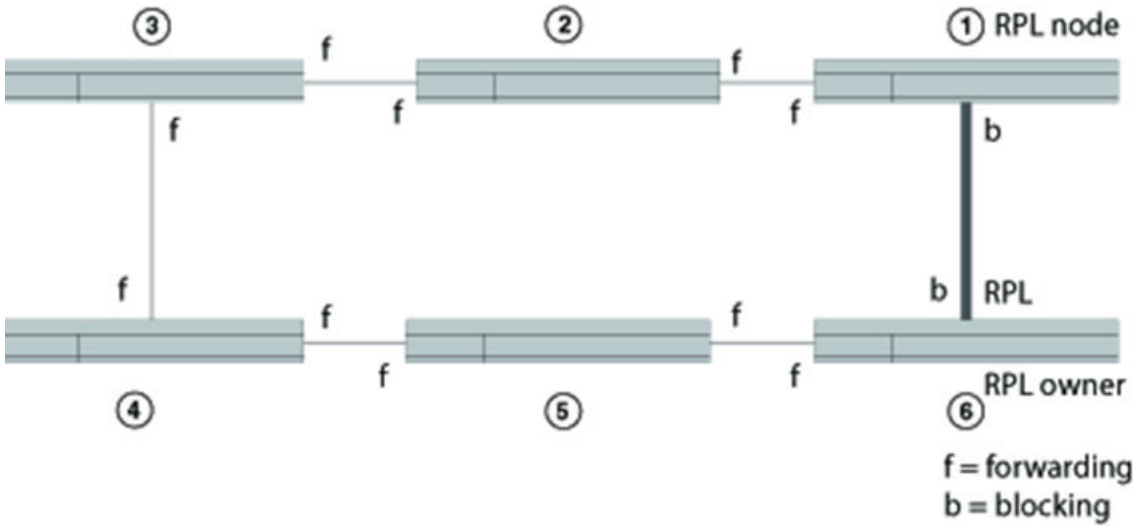
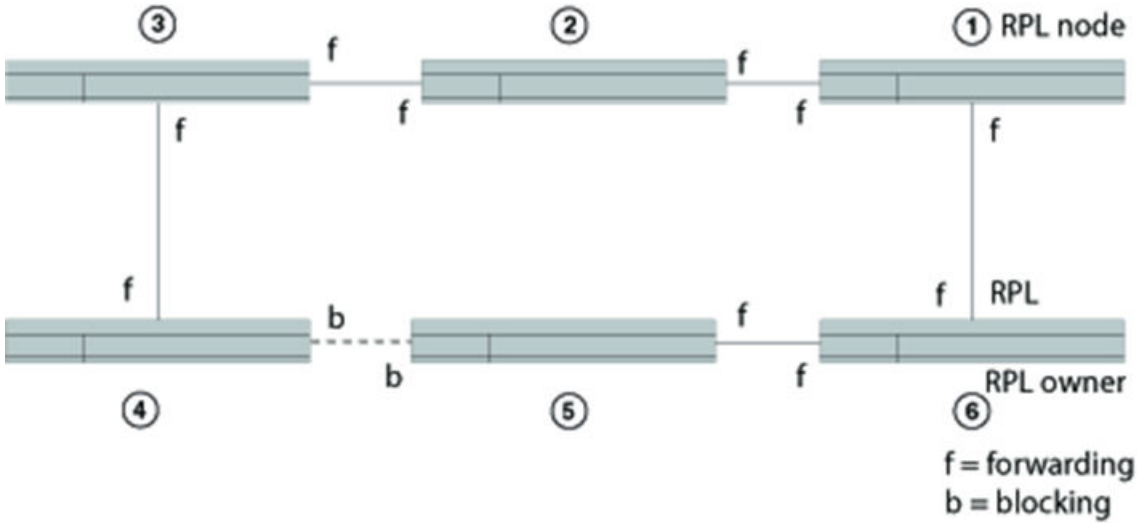


Figure 139 shows the same Ethernet ring topology after a failure at the forwarding port of ERN 4 when a signal fail triggered, and ring protection was needed. ERN 6 unblocked the RPL port and the RPL node changed the blocking port to the forwarding state.

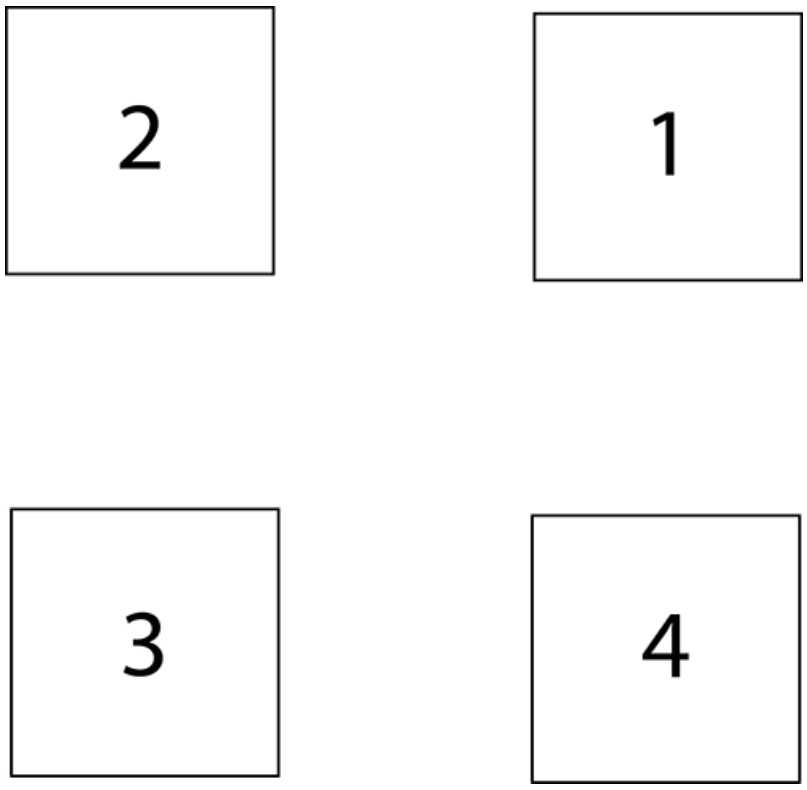
FIGURE 136 ERP topology in a Protected state



Manual switch

In the absence of a failure, an operator-initiated manual switch (MS) moves the blocking role of the RPL by blocking a different ring link and initiates the node sending a R-APS (MS) to inform the RPL owner to unblock the RPL. This can occur if no higher priority request exists in the ring. See Figure 140. The thick line between ERNs 1 and 2 indicate that dual-end blocking is enabled.

FIGURE 137 Manual Switch example



The node, which receives the R-APS (MS), forwards it to the adjacent nodes. If the receiving node is already in the Idle or Pending state, it unblocks the non-failed port and stops transmitting R-APS messages. Only one MS can exist in the topology at any time. An MS condition has to be manually cleared with the **no** command.

> **NOTE**
> If any ERN is in an FS state or in a protected state through an SF event and an operator tries to configure an MS, the ERN will reject the request.

When a manual switch is cleared by an operator on the same node on which the MS is configured, the node keeps the port in a blocking state, sends out a R-APS (NR) to the adjacent node, and starts the guard timer. Other nodes that receive the R-APS (NR) forward the message. When the RPL owner receives this message, then the RPL owner starts the WTR timer. When the WTR timer expires, the RPL owner sends out a R-APS (NR, RB), blocks the RPL, and flushes the FDB. Other nodes in the topology that receive the R-APS (NR, RB) unblock any non-failed port and flush the FDB.

Figure 140 shows a manual switch on ERN 3, which is a non-RPL node. In order to clear the MS condition, the operator must enter the manual switch command from ERN 3. The sequence of messages and actions is as shown in Manual switch.

MS on Non-RPL node

Non-RPL node with error (ERN 3)	RPL owner (ERN 1) and RPL node (ERN2)	Other Non-RPL node (ERN 4)
From the Idle state, ERN3: 1. Blocks the MS port 2. Sends the RAPS (MS)		

Non-RPL node with error (ERN 3)	RPL owner (ERN 1) and RPL node (ERN2)	Other Non-RPL node (ERN 4)
3. Flushes the FDB 4. Enters the manual switch (MS) state		
		From the Idle state, ERN 4: 1. Forward R-APS (MS) 2. Flush the FDB 3. Enter the MS state
	From the Idle state, ERN 1: 1. Forwards R-APS (MS) 2. Unblocks the RPL 3. Flushes the FDB 4. Enters the MS state	

After the manual switch is triggered, the operator can clear it with the **no** command and MS recovery will begin. Manual switch shows the sequence of events during the MS recovery process.

MS recovery process

Non-RPL node with error (ERN 3)	RPL owner (ERN 1)	RPL node (ERN2) with dual-end blocking enabled	Non-RPL node (ERN 4)
From the MS state, ERN 3: 1. Stops sending R-APS (MS) 2. Sends R-APS (NR) 3. Continues to block the port 4. Enters the Pending state			
	From the MS state, ERN 1: 1. Receives the R-APS (NR) 2. Starts the WTB timer 3. Forwards the R-APS (NR) 4. Enters the Pending state 5. After the WTB timer expires, blocks the RPL 6. Flushes the FDB 7. Sends R-APS (NR, RB) 8. Enters the Idle state	From the MS state, ERN 2: 1. Receives the R-APS (NR) 2. Forwards the R-APS (NR) 3. Enters the Pending state	From the MS state, ERN 2: 1. Receives the R-APS (NR) 2. Forwards the R-APS (NR) 3. Enters the Pending state
From the Pending state, ERN 3: 5. Receives the R-APS (NR, RB) and unblocks the blocking port 6. Forwards the R-APS (NR, RB) 7. Flushes the FDB 8. Enters the Idle state		From the Pending state, ERN 2: 4. Blocks the RPL 5. Forwards the R-APS (NR, RB) 6. Flushes the FDB 7. Enters the Idle state	From the Pending state, ERN 4: 4. Forwards the R-APS (NR, RB) 5. Flushes the FDB 6. Enters the Idle state

Forced switch

Forced switch (FS) is an operator-initiated mechanism that moves the blocking role of the RPL to a different ring link followed by unblocking the RPL, even if one or more failed links exist in the ring.

The node configured to initiate an FS blocks the port and sends out a R-APS (FS) to inform other nodes to unblock any blocked ports (including failed ones) as long as no other local request with higher priority exists. The RPL owner unblocks the RPL and flushes the FDB.

Any node accepting a R-APS (FS) message stops transmitting R-APS messages.

Multiple FS instances can be configured in the topology even when the topology is in the same segment where an FS is being cleared by **no** command. When an operator clears an FS on the same node where an FS is configured, this node keeps the port in the blocking state, sends out a R-APS (NR) to adjacent nodes, and starts the guard timer. Other nodes that receive the R-APS (NR) forward the message. When the RPL owner receives this message, the RPL owner starts the WTB timer. When the WTB timer expires, the RPL owner sends out a R-APS (NR, RB), blocks the RPL, and flushes the FDB. Other nodes in the topology that receive the R-APS (NR, RB) unblock any non-failed port and flush the FDB.

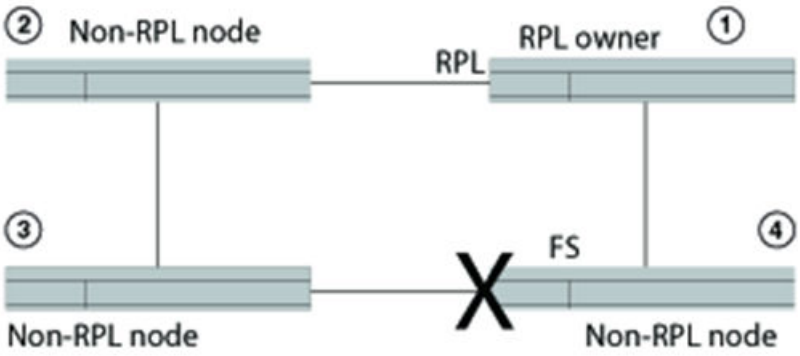
An FS request can be accepted no matter what state the topology is in. Since the local FS and R-APS (FS) are higher priority than SF; an SF occur i ng later than FS will not trigger the SF process. In addition, because the local FS and R-APS (FS) are higher priority than SF, when a node receives a R-APS (FS) without any local higher priority event, it will unblock any blocked port. The node with the failed link also unblocks the blocked port; but because the link has failed, the topology is broken into segments.

Since the local FS and R-APS (FS) are higher priority than a local SF clear when the link failure is removed without any local higher priority event, the nodes with the recovering link do not trigger SF recovery.

After the operator clears the FS condition on the node, the node starts the guard timer and sends out a R-APS (NR). When the RPL owner receives a R-APS(NR), it stops the WTB timer and starts the guard timer. The RPL owner blocks the RPL and sends out a R-APS (NR, RB). Any node receiving a R-APS (NR, RB) unblocks the non-failed blocked port. If the guard timer is still running on the node with previous FS, this node ignores R-APS messages until the guard timer expires. The topology is again broken into segments. After this node processes the R-APS (NR, RB), however, it unblocks the blocked node; and the topology is in a loop free state and in one segment.

Figure 141 shows a port failure on ERN 4.

FIGURE 138 Single forced switch scenario



Forced switch shows the sequential order of events triggered as a result of an operator-initiated forced switch command entered from ERN 4.

Single FS process--operator entered the forced switch command from ERN 4

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
Idle	Idle	Idle	From the Idle state, ERN 4: 1. Processes the Forced Switch command 2. Blocks the requested port 3. Transmits R-APS (FS) 4. Unblocks the non-requested port 5. Flushes the FDB 6. Enters the Forced Switch (FS) state
From the Idle state, ERN 1: 1. Unblocks the RPL 2. Flushes the FDB for first time 3. Forwards R-APS(FS) 4. Enters the FS state	From the Idle state, ERN 2: 1. Unblocks the port 2. Flushes the FDB for the first time 3. Forwards R-APS(FS) 4. Enters the FS state	From the Idle state, ERN 3: 1. Unblocks the port 2. Flushes the FDB for the first time 3. Forwards R-APS(FS) 4. Enters the FS state	
From the FS state, ERN 1 forwards R-APS	From the FS state, ERN 2 forwards R-APS	From the FS state, ERN 3 forwards R-APS	From the FS state, ERN 4: 7. Transmits R-APS(FS) 8. Terminates the received R-APS on the blocking port 9. Terminates its own R-APS(FS)
All ERNs remain in FS state.			

Next, the operator enters the **no** command to clear the forced switch. For this example, the operator initiated the forced switch from ERN 4 and must clear it from ERN 4. Forced switch shows the forced switch recovery process in sequential order.

FS clear process

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
			From the FS state, ERN 4: 1. Starts the guard timer 2. Stops transmitting R-APS(FS 3. Transmits R-APS(NR) 4. Keeps blocking the port 5. Enters Pending state
From FS state, ERN 1: 1. Forwards R-APS 2. Starts the guard timer 3. Starts the WTB timer 4. Enters Pending state			
	From FS state, ERN 2: 1. Forwards R-APS	From FS state, ERN 3: 1. Forwards R-APS	

RPL owner (ERN1)	Non-RPL node (ERN 2)	Non-RPL node (ERN 3)	Non-RPL node (ERN 4)
	2. Starts the guard timer 3. Enters the Pending state	2. Starts the guard timer 3. Enters the Pending state	
After the WTB timer expires, from the Pending state ERN 1: 5. Blocks the RPL port 6. Transmits R-APS(NR,RB) 7. Unblocks the non-RPL port 8. Flushes the FDB 9. Enters the Idle state			
	From the Pending state, ERN 2: 4. Flushes the FDB 5. Forwards R-APS(NR,RB) 6. Enters the Idle state	From the Pending state, ERN 3: 4. Stops transmitting R-APS 5. Unblocks ports 6. Flushes the FDB 7. Forwards R-APS(NR,RB) Enters the idle state	From the Pending state, ERN 4: 6. Stops transmitting R-APS 7. Unblocks ports 8. Flushes the FDB 9. Forwards R-APS(NR,RB) 10. Enters the Idle state
From the idle state, ERN 1: 10. Receives its own R-APS(NR,RB) 11. Stops transmitting R-APS 12. Remains in the Idle state			

Double Forced Switch

A local FS is of a higher priority than a received R-APS (FS); therefore, the local FS request blocks the port even when the node receives a R-APS(FS) from another FS request of another node.

After the first FS clears, the node starts the guard timer and sends out a R-APS (NR). The adjacent nodes of the first cleared FS node will not process or forward the R-APS (NR) because they are still receiving R-APS (FS) from the second FS node. When the first FS node receives R-APS (FS) from the second FS nodes, it unblocks any blocked port and stops transmitting any lower priority R-APS messages. At this point, the topology follows the single FS process, as previously described.

Dual-end blocking

Dual-end blocking is a user configurable feature to directly conserve bandwidth of the RPL and indirectly conserve processing power of the RPL owner. When you configure a node in a major ring adjacent to the RPL owner to be an RPL node with dual-end blocking enabled, data traffic and R-APS messages will not be forwarded to the blocked port of the RPL owner.

When a failure occurs in the ring and the RPL node (not the RPL owner) receives a R-APS (of type SF, FS, or MS), the RPL node unblocks the configured dual-end blocked port. When the RPL node receives a R-APS (NR, RB), it reblocks the originally configured dual-end blocked port. To configure dual-end blocking you need to configure the RPL and dual-end blocking on both the RPL owner and the adjacent peer (RPL node).

Non-revertive mode

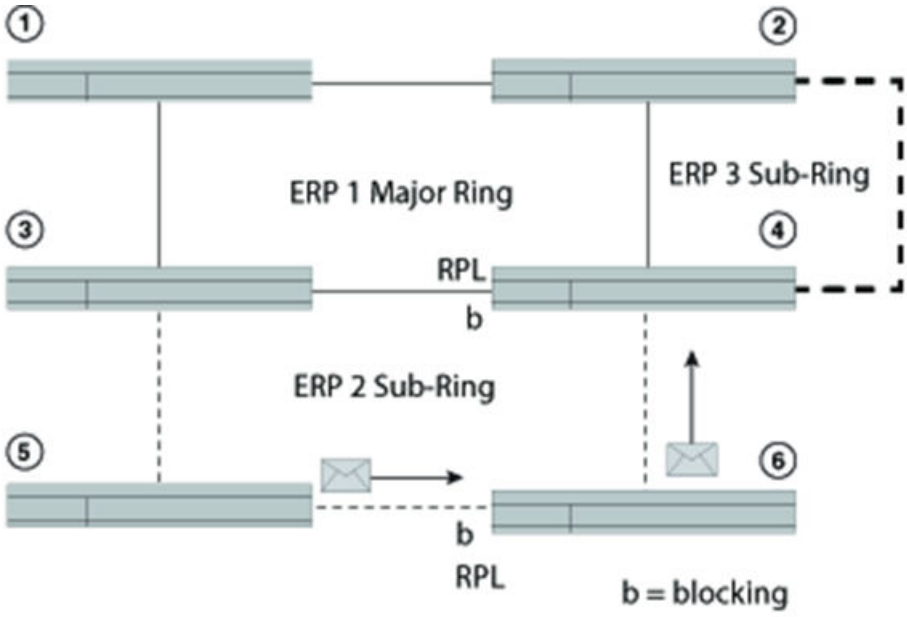
In non-revertive mode, the traffic channel is allowed to use the RPL, if it is not failed, after a switch condition clears. In the recovery from a Protection state, the RPL owner generates no response regarding the reception of NR messages. When other healthy nodes receive the NR message, there is no action in response to the message. After the operator issues a **no** command for non-revertive mode at the RPL owner, the non-revertive operation is cleared, WTB or WTR timer starts, as appropriate, and the RPL owner blocks its port to the RPL and transmits a R-APS (NR, RB) message. Upon receiving the R-APS (NR, RB), any blocking node should unblock its non-failed port.

Interconnected rings

Interconnected rings consist of one major ring and one or more sub-rings with shared physical links. The ring links between the interconnection nodes are controlled and protected by the ERP ring to which they belong. A sub-ring is similar to the major ring in that each sub-ring has an RPL and an RPL owner. The RPL owner can be configured in any node belonging to the ring.

The dotted lines in Figure 142 show two of the many potential sub-rings that you can configure.

FIGURE 139 Interconnected rings with major and sub rings shown



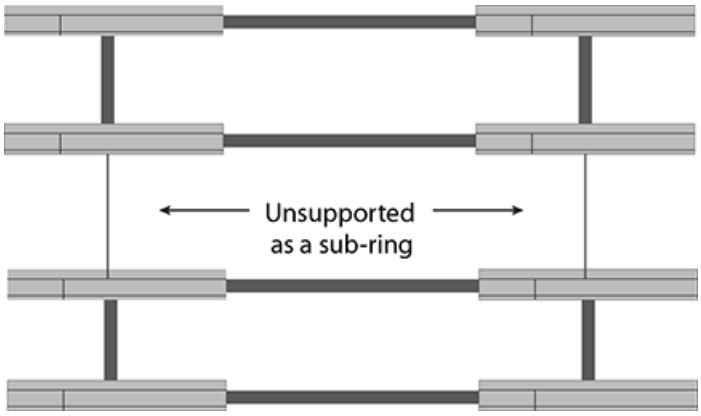
When a sub-ring initializes, each ERN in the non-closed ERP sends out a R-APS (NR). After the RPL owner receives a R-APS (NR), it blocks the RPL; and the RPL owner sends out a R-APS (NR, RB). The shared link remains blocked even if the shared link has a SF error. The blocking state in ERP means the R-APS channel is blocked at the same port where the traffic channel is blocked, except on sub-rings without use of R-APS virtual channel.

A sub-ring in segments interconnecting major rings is not supported. Figure 143 shows a major ring and two segments not supported as a sub-ring.

Blocking prevents R-APS messages received at one ring port from being forwarded to the other ring port; it does not prevent the R-APS messages locally generated at the ERP control process from being transmitted over both ring ports, and it also allows R-APS messages received at each port to be delivered to the ERP control process.

Each ERN in a major ring terminates R-APS messages received on a blocking port and does not forward the message if the port is in a blocking state. Each ERN in a sub-ring, however, still forwards the R-APS messages received on a blocking port.

FIGURE 140 Unsupported sub-ring in segments



FDB flush optimization

The FDB stores the node ID and BPR sent in the R-APS messages. When an ERN receives a new R-APS message, it compares the received node ID and BPR to the node ID and BPR in its memory. If the pair vary from the previously stored pair, the ERN deletes the previous pair and stores the new pair. The device then triggers a FDB flush unless the DNF (No Not Flush) is set in the message.

FDB optimization is achieved with the following features:

- Non-revertive mode alleviates the need to flush the FDB after a link failure with link protection (SF) condition
- Dual-end blocking decreases attempted messages and traffic to the RPL blocking port
- Interconnected ring support to decrease the latency for messaging
- Do Not Flush (DNF) messages

Configuring ERP

To configure and initialize ERP using only APS you must set up one RPL owner and one or more Non-RPL nodes. The minimum configuration tasks are listed in this section.

Before configuring ERP, however, you must have already configured a VLAN and ports.

> NOTE
> ERP only supports topology-groups if the ERP interfaces are in the same VLAN

You must perform the following minimum configuration tasks for the RPL owner:

- Configure an ERP instance
- Set the left and right interfaces

- Set the role as owner
- Set the RPL
- Enable the configuration

You must perform the following minimum configuration tasks for each non-RPL node:

- Configure an ERP instance
- Set the left and right interfaces
- Enable the configuration

Sample configuration

The following example is of an ERP configuration consisting of four devices: an RPL owner, an RPL node, and two non-RPL nodes.

NOTE
Before configuring any ERP settings, configure the VLAN and ports.

Device 1 RPL owner

NOTE
Optionally, you can configure the non-revertive mode feature. This setting can only be set on the RPL owner.

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#rpl vlan 2 e1/2
(config-erp-1)#rpl-owner
(config-erp-1)#enable
```

Device 2 RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#rpl vlan 2 e 1/1
(config-erp-1)#enable
```

Device 3 Non-RPL node

```
(config)# erp 1
(config-erp-1)#right-interface vlan 2 e 1/2
(config-erp-1)#left-interface vlan 2 e 1/1
(config-erp-1)#enable
```

Device 4 Non-RPL node

(config)# erp 1

(config-erp-1)#right-interface vlan 2 e 1/2

(config-erp-1)#left-interface vlan 2 e 1/1

(config-erp-1)#enable

Configuring ERP with IEEE 802.1ag

To configure and initialize ERP using APS and IEEE 802.1ag you must set up one RPL owner and one or more Non-RPL nodes. Other nonparticipating switches can exist in the ring.

You must perform the following minimum configuration tasks for the RPL owner:

- Configure an ERP instance
- Set the left and right interfaces
- Set the role as owner
- Set the RPL
- Enable the configuration

You must perform the following minimum configuration tasks for each non-RPL node:

- Configure an ERP instance
- Set the left and right interfaces
- Configure the maintenance entity group end points (MEP) from each ERN, which can have a role of RPL owner or non-RPL node, adjacent to switches not participating in the ERP configuration
- Enable the configuration

ERP commands

This section lists ERP configuration commands.

Assigning ERP IDs

You must assign an ERP ID. This ID number is used to:

- Filter and clear statistics associated with a particular ERP ID
- Delete the non-revertive mode in the case of an RPL owner
- Clear WTR and WTB timers

The *erp_id* value is a number from 1 to 255.

Syntax: erp *erp_id*

For example, to assign the number 10 to the ERP, enter:

```
(config)# erp 10
```

Naming an Ethernet Ring Node

From within the ERP configuration shell, you can optionally name an ERN with a meaningful name. The name must be 31 alphanumeric characters or fewer; and the name can use the "underscore" and "dash" special characters.

Syntax: [no] name *erp_name*

For example, to assign the name "to_brocade1" to an ERN with ID number 10, enter:

device `(config)# erp 10`

device `(config-erp-10)# name "to_brocade1"`

Use the **no** command to remove the name.

Configuring the default MAC ID

You can configure the MAC ID. The device appends this ID number to the end of the permanent portion of the ERP MAC address (00-00-00-00-00- <01 or ERP ID>) in R-APS messages. By default 00-00-00-00-00-01 is used as the dst MAC, which is always used by Version 1 of ITU-T 8032. If Version 2 is configured, then the **raps-default-mac** command can be negated by entering the **no raps-default-mac** command. The configured ERP ID will appear as the last 8-bit number in the destination MAC.

For more information about feature support for version 1 and 2, see Setting the ITU-T G.8032 version number on page 512.

Syntax: [**no**] **raps-default-mac**

Configuring R-APS MEL value

The R-APS Maintenance Entity Group Level (MEL) value can be configured. The R-APS MEL value is carried in ERP PDUs. The default R-APS MEL value is 7.

Syntax: [**no**] **raps-mel mel value**

Configuring R-APS topology change propagation

When there is a topology change in a sub-ring, the information needs to be propagated over the major ring. This propagation involves transmission of RAPS (MAC flush event) PDUs over the major ring associated with the sub-ring. This results in a filter database (FDB) flush on the major ring nodes.

Syntax: [**no**] **raps-propagate-tc**

Enabling the ERP configuration

You must apply the **enable** command to activate an ERP configuration. You can use the **no** command to disable the configuration.

Within an interconnected ring topology, in the major ring, you must first configure two interfaces. In a sub-ring, at least one interface must first be configured before enabling the ERP instance.

Syntax: [no] enable

Example of a non-RPL node configuration in a major ring:

(config)# erp 1

(config-erp-1)#right-interface vlan 2 e 1/2

(config-erp-1)#left-interface vlan 2 e 1/1

(config-erp-1)#enable

Configuring interfaces

Each ERN in a major ring must have explicitly defined left and right interfaces so that ERP can function properly. ERNs in a sub-ring must have at least one interface defined so that ERP can function properly.

For proper operation you must configure the interfaces following the same manner on each ERN, such as left/ right, left/ right, and so on.

Syntax: [no] left-interface [**vlan** *vlan-id* | **esi esi_name vlan vlan_id**] **e** *slot/ number*

Syntax: [no] right-interface [**vlan** *vlan-id* | **esi esi_name vlan vlan_id**] **e** *slot/ number*]

Use the **no** command to remove the configuration of each interface.

Assigning the RPL owner role and setting the RPL

Each ring needs to have one RPL owner for each ring. The RPL owner's role is to block traffic on one port when no failure exists in the ring. The blocked port will be the left interface that you initially configured. After configuring the ERN to be the RPL owner, you next must set the RPL. To set the RPL you need to specify the VLAN and Ethernet slot and port.

> NOTE
> When you assign the role of RPL owner, you must also configure the RPL.

Syntax: [no] rpl-owner

Syntax: [no] rpl [**vlan** *vlan-id* | **esi esi_name vlan vlan_id**] **e** *slot/number*

Enabling sub-rings for multi-ring and ladder topologies

In multi-ring and ladder topologies, you can enable the multi-ring feature.

Interconnected rings consists of one major ring and at least one sub-ring within the same VLAN or different VLANs. A sub-ring is not a complete ring. Nodes within a sub-ring can be configured as a one-arm ring. Each sub-ring must have its own RPL owner and RPL ports as appropriate.

RPL ports and the RPL owner also need to be configured in a sub-ring. All ERP features are available in both major and sub-rings.

R-APS PDUs only flow in the nodes with same ring ID. The R-APS PDU can be forwarded through the port in sub-ring blocking state.

Syntax: [no] sub-ring [**parent-ring-id erp-id**]

The **parent-ring-id** is used when there are different VLANs on the major ring and sub-ring. In such a case, use the parent-ring-id configuration to determine the ring to which the sub-ring is connected. The parent ring can be either another sub-ring or major ring connected to the sub-ring.

Use the **no** command to delete the sub-ring support.

If you have six nodes you can put them in one ring. The latency time for packet transport, however, increases in big topologies even within the same VLAN, so it is better to separate them out.

Achieving sub-50ms ring protection switch time

The G.8032v2 ERP implementation has been enhanced to achieve sub-50ms ring protection switch time. This enhancement involves optimizations to reduce the number of MAC flushes, temporary flooding of traffic while MAC flush is in progress, and faster link failure detection using Connectivity Fault Management (CFM).

You will need to configure the following to achieve sub-50ms ring protection switch time.

Enter the following command to allow temporary flooding of traffic during MAC flush. Use the **no** version of this command to disable the flooding of traffic.

```
Brocade(config-erp-1)#flooding-enable
```

Syntax: [no] flooding-enable

During topology change, there are multiple MAC flushes triggered by ERP protocol. Optimizations have been made to ERP protocol to reduce the number of MAC flushes to achieve faster convergence. These optimization can be enabled using the **fdb-flush-optimization** command. Use the **no** version of this command to disable the flush optimization.

```
Brocade(config-erp-1)#fdb-flush-optimization
```

Syntax: [no] fdb-flush-optimization

IEEE 802.1ag can be used to monitor the ERP interfaces for signal failures.The **dot1ag-compliance** command allows MD and MA's configured as part of IEEE 802.1ag to be associated with an ERP instance. Use the **no** version of this command to disable the **dot1ag-compliance** command.

```
Brocade(config-erp-1)#dot1ag-compliance domain-name erp ma-name ma-erp
```

Syntax: [no] dot1ag-compliance domain-name *domain-name* **ma-name** *ma-name*

The **domain-name** parameter specifies the maintenance domain name for 802.1ag CFM.

The **ma-name** parameter specifies the maintenance association name. This can be up to 21 characters long.

Configuration recommendations for dot1ag:

- Dot1ag configuration for ERP is recommended only for copper ports as link failure detection time is higher when compared to fiber ports.
- If dot1ag is configured, it is recommended to ensure broadcast traffic in the network does not cross 100% of link bandwidth utilization. If the broadcast traffic crosses 100% of the link bandwidth utilization, then CCM packets might be lost which can result in MEPs moving to failed state causing ERP state fluctuations.

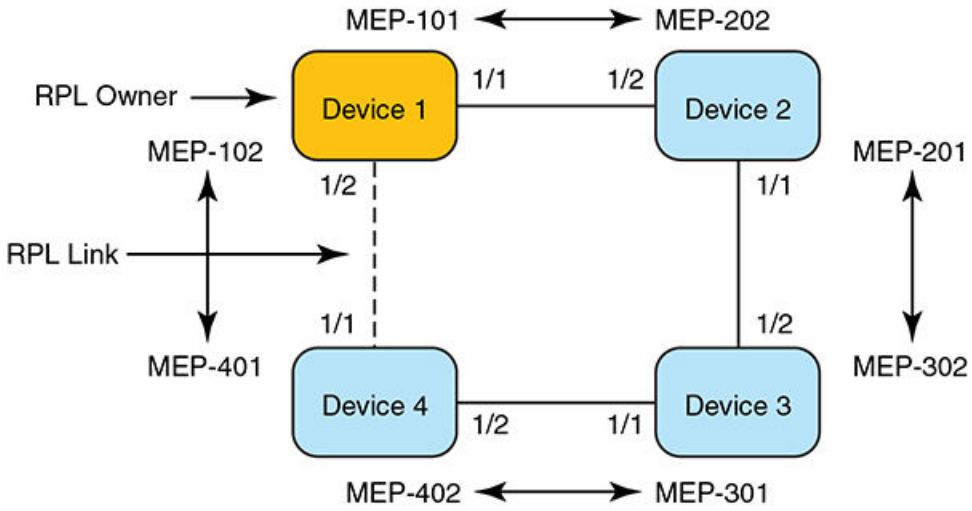
  NOTE
  Sub-50ms convergence time may not be achievable with LAG interfaces.

This command needs to be enabled under the link MA configuration of IEEE 802.1ag. This allows configuring MEPs to individual links of the LAG, and enables monitoring of each member link of that LAG. Use the **no** version of this command to disable the individual link monitoring.

```
Brocade(config-cfm-md-erp-ma-ma-erp)#individual-link-monitoring
```

Syntax: [no] individual-link-monitoring

FIGURE 141 Network diagram for ERP



Configuration example

NOTE
The VLAN CPU protection needs to be enabled on the VLANs.

Device 1 Configuration steps

CFM configuration:

```
device#configure terminal
device(config)#cfm-enable
device(config-cfm)#domain-name erp id 1 level 1
device(config-cfm-md-erp)#ma-name ma-erp link-ma priority 7
device(config-cfm-md-erp-ma-ma-erp)#individual-link-monitoring
device(config-cfm-md-erp-ma-ma-erp)#mep 101 down port eth 1/1
device(config-cfm-md-erp-ma-ma-erp)#mep 102 down port eth 1/2
```

VLAN Configuration:

```
device(config)#vlan 100
device(config-vlan-100)#tag eth 1/1 eth 1/2
device(config-vlan-100)#vlan-cpu-protection
```

ERP Configuration:

```
device(config)#erp 1
device(config-erp-1)#left-interface vlan 100 eth 1/1
device(config-erp-1)#right-interface vlan 100 eth 1/2
device(config-erp-1)#rpl-owner
device(config-erp-1)#rpl vlan 100 eth 1/2
device(config-erp-1)#flooding-enable
device(config-erp-1)#fdb-flush-optimization
device(config-erp-1)#dot1ag-compliance domain-name erp ma-name ma-erp
device(config-erp-1)#enable
```

Device 2 Configuration steps

CFM Configuration:

```
device#configure terminal
device(config)#cfm-enable
device(config-cfm)#domain-name erp id 1 level 1
device(config-cfm-md-erp)#ma-name ma-erp link-ma priority 7
device(config-cfm-md-erp-ma-ma-erp)#individual-link-monitoring
device(config-cfm-md-erp-ma-ma-erp)#mep 201 down port eth 1/1
device(config-cfm-md-erp-ma-ma-erp)#mep 202 down port eth 1/2
```

VLAN Configuration:

```
device(config)#vlan 100
device(config-vlan-100)#tag eth 1/1 eth 1/2
device(config-vlan-100)#vlan-cpu-protection
```

ERP Configuration:

```
device(config)#erp 1
device(config-erp-1)#left-interface vlan 100 eth 1/1
device(config-erp-1)#right-interface vlan 100 eth 1/2
device(config-erp-1)#flooding-enable
device(config-erp-1)#fdb-flush-optimization
device(config-erp-1)#dot1ag-compliance domain-name erp ma-name ma-erp
device(config-erp-1)#enable
```

Configuring non-revertive mode

After the Ethernet Ring enters a protected state, if you do not want the topology to return to the original state you can use the **non-revertive-mode** command to keep it in the new state. Enter this command on the RPL owner only, and then enter the **enable** command.

Syntax: [no] non-revertive-mode

Use the **no** command to remove the non-revertive mode setting.

Configuring and clearing a forced switch

An operator can use the forced switch (FS) mechanism when no errors, a single error, or multiple errors are present in the topology. You can enter this command multiple times. You need to explicitly specify the VLAN and Ethernet slot and port.

Syntax: [no] forced-switch [**vlan** *vlan-id* | **esi** *esi_name* **vlan** *vlan_id*] **e** *slot/port*]

Use the **no forced-switch** command to remove the forced switch mechanism.

Configuring and clearing a manual switch

Manual switch (MS) is an operator-initiated process that manually blocks a desired port in a ring. You need to explicitly specify the VLAN, Ethernet slot, and port from the desired device.

Syntax: [no] manual-switch [**vlan** *vlan* | **esi** *esi_name* **vlan** *vlan_id*] **e** *slot/port*]

Use the **no manual-switch** command to remove the manual switch mechanism.

Configuring dual-end blocking

You can configure dual-end blocking to optimize your ERP configuration. The RPL node must be adjacent to the RPL owner.

When you configure the RPL on an ERN that is adjacent to the RPL owner, you are enabling the dual-end blocking feature and changing the ERN's role to that of RPL node. You configure the RPL node with the **rpl** command. Before configuring dual-end blocking, you must

verify that the RPL node is actually the correct peer and obtain the RPL link settings; an incorrect setting will cause incorrect port blocking.

> NOTE
> The RPL node must be a peer of the RPL owner, and the RPL must be configured on this peer; otherwise, the device will perform incorrect port blocking behavior.

Syntax: [no] rpl [**vlan** *vlan-id* | **esi** *esi_name* **vlan** *vlan_id slot/number*]

Configuring the guard timer

The guard timer prevents ERNs from acting upon outdated R-APS messages and prevents the possibility of forming a closed loop. The guard timer enforces a period during which an ERP topology ignores received R-APS.

This timer period should always be greater than the maximum expected forwarding delay in which an R-APS message traverses the entire ring. The longer the period of the guard timer, the longer an ERN is unaware of new or existing relevant requests transmitted from other ERN and, therefore, unable to react to them.

The guard timer is used in every ERN, once a guard timer is started, it expires by itself. While the guard timer is running, any received R-APS request/state and Status information is blocked and not forwarded to the priority logic. When the guard timer is not running, the R-APS request/state and status information is forwarded unchanged.

> NOTE
> The ITU-T G.8032 standard defines the guard timer period as configurable in 10 ms increments from 10 ms to 2000 ms (2 seconds) with a default value of 500 ms.

The guard timer is activated when an ERN receives an indication that a local switching request, such as a clear signal fail, manual switch, or forced switch, is cleared.

The guard timer can be configured in 100 ms increments from 1200 ms to 4000 ms (4 seconds); the default value is 1500 ms (1.5 seconds). The guard timer cannot be stopped manually.

Syntax: guard-time *time-value*

Configuring and clearing the wait to restore timer

For SF recovery situations, you can configure the wait to restore (WTR) timer on the RPL owner to prevent frequent operation of the protection switching due to the detection of intermittent signal failures. When recovering from a Signal Failure, the WTR timer must be long enough to allow the recovering network to become stable.

This WTR timer is activated on the RPL Owner Node. When the relevant delay timer expires, the RPL owner initiates the reversion process by transmitting an R-APS (NR, RB) message. The WTR timer is deactivated when any higher priority request preempts this timer. The WTR timers may be started and stopped. A request to start running the WTR timer does not restart the WTR timer. A request to stop the WTR timer stops the WTR timer and resets its value. The Clear command can be used to stop the WTR timer. While WTR timer is running, the WTR running signal is continuously generated. After the WTR timer expires, the WTR running signal is stopped, and the WTR Expires signal is generated. When the WTR timer is stopped by the clear command, the WTR Expires signal is not generated.

When configured, the RPL owner waits until the timer expires before transmitting the R-APS(NR,RB) message to initiate the reversion process. While the timer is in effect, the WTR running signal is continuously generated. You can configure the WTR timer in 1 minute increments from 1 to 12 minutes; the default value is 5 minutes.

This timer can be stopped by issuing the **clear erp** command.

Syntax: wtr-time *time-value*

Testing the WTR timer

You can enter the **fast-wtr-time** command to test your configuration. Instead of having to wait 5 minutes for the timer to expire, you wait 5 seconds. This command changes the timer's unit of measure from minutes to seconds.

Syntax: [no] fast-wtr-time

Use the **no** command to return the unit of measure to minutes.

Configuring and clearing the WTB timer

The WTB timer ensures that clearing of a single FS command does not trigger the reblocking of the RPL when multiple FS situations co-exist in an Ethernet Ring. When recovering from an MS or FS command, the delay timer must be long enough to receive any latent remote FS or MS.

While it is running, the WTB running signal is continuously generated. The WTB timer is 5000ms (5 seconds) longer than the guard timer. You can configure this timer in 100 ms increments from 5100ms to 7000ms (7 seconds); the default value is 5500ms.

The WTB timer can be stopped through the CLI by entering the **clear erp***erp_id***wtb-timer** command.

Syntax: wtb-time *time-value*

Configuring a hold-off timer

The hold-off timer is used in each ERN to prevent unnecessary Signal Fail events due to port flapping. If you configure a non-zero hold-off timer value, when a link error occurs, the event will not be reported immediately. When the hold-off timer expires, ERP checks if the error still exists.

The hold-off timer is used in every ERN. When a new defect occurs (new SF), this event will not be reported immediately to trigger protection switching if the provisioned hold-off timer value is non-zero. Instead, the hold-off timer is started. When the hold-off timer expires, the trail that started the timer is checked as to whether a defect still exists. If one does exist, that defect is reported and protection switching is triggered.

You can configure the hold-off timer in 100ms increments from 0 to 10,000 ms (10 seconds); the default value is 0 ms. The hold-off timer value cannot be stopped through the CLI.

Syntax: holdoff-time *time-value*

Configuring the message interval time

The message interval time of R-APS messages continuously sent within an ERP ring can be configured. You can configure the interval in 100ms increments from 100ms to 5000ms (5 seconds); the default value is 5000ms.

Syntax: message-interval *time-value*

Setting the ITU-T G.8032 version number

You can configure the ERP configuration to use G.8032 version 1 or 2. The default value is version 2. Setting the ITU-T G.8032 version number lists the feature and MAC ID differences between versions 1 and 2.

NOTE
The ERP **version** command does not have a shortened form. You must enter the complete command.

1. Signal Fail Signal Fail recovery
2. Always uses 01:19:A7:00:00:01 as the ERP ID in R-APS messages

3. Signal Fail Signal Fail recovery Manual Switch Forced Switch Non-revertive Interconnected rings RPL configuration on non-RPL owner
4. Allows use of the ERP ID for the last two bytes of the MAC ID (01:19:A7:00:00:erp-id)

Syntax: version *version_number*

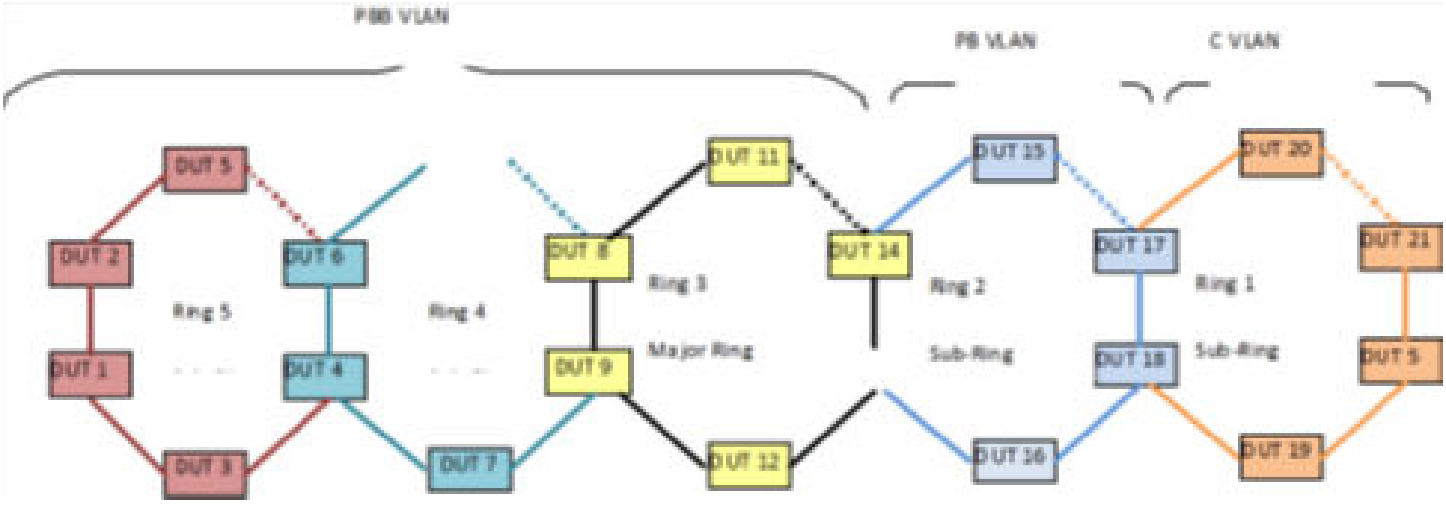
You can view the version by entering the **show erp** command. The version appears on the top line directly after the ERP ID.

ERP over ESI VLAN (Brocade NetIron CES Series and Brocade NetIron CER Series)

Figure 145 shows a diagram of one of the sample topologies that is used to explain deployment of ERP over PBB using the ESI model. In the diagram, Ring 3 is the major ring while all others are sub-rings. Each ESI VLAN will be running a separate instance of ERP. Any ring can act as a major ring. However, it is recommended that the B-VLAN be used as a part of Major ring.

When a network involves PB and PBB rings, one of the PBB ring must be configured as a major ring. In the case of a PB only network, a PB ring can be the major ring. There can be only one major ring in a given network.

FIGURE 142 ERP configuration over ESI network



In general scenarios of ERP, a multiple VLANs span across multiple rings forming a major ring and sub-rings. The major ring and sub-ring is determined by the ERP configuration and the ERP protocol operates keeping in focus the traffic flow.

Interconnection rings with different VLANs

In the ESI model, traffic flow is determined by the mapping of one ESI VLAN relationship to another ESI VLAN. In ERP over ESI VLAN (Brocade NetIron CES Series and Brocade NetIron CER Series) on page 513, the ESI CVLAN1 is a client for the ESI SVLAN1 instance. So traffic from Ring 1(CVLAN1) gets encapsulated with an S-TAG (SVLAN1) in Ring 2. Similarly, ESI SVLAN1 is a client for the ESI BVLAN1 instance, resulting in traffic from Ring 2 encapsulated with a PBB header in Ring 3. The traffic forwarding from Ring 3 towards Ring 4 and Ring 5 occurs as in any normal VLAN as all the rings are running on same VLAN (BVLAN1).

To support the ERP over ESI model, each ring needs to run its own ERP instance with an ESI VLAN. For Ring 1, the ERP instance needs to run on CVLAN1, for Ring 2 on SVLAN1 and for Rings 3, 4 and 5 on BVLAN1. Some of the nodes, such as DUT17 and DUT18, connect one ERP ring to another ERP ring. These are called interconnection nodes. Each interconnection node in ERP over ESI VLAN (Brocade NetIron CES Series and Brocade NetIron CER Series) on page 513, runs two instances of ERP. For example, in the case of DUT17 one ERP instance is run for Ring 1 on CVLAN1 and another ERP instance for Ring 2 on SVLAN1.

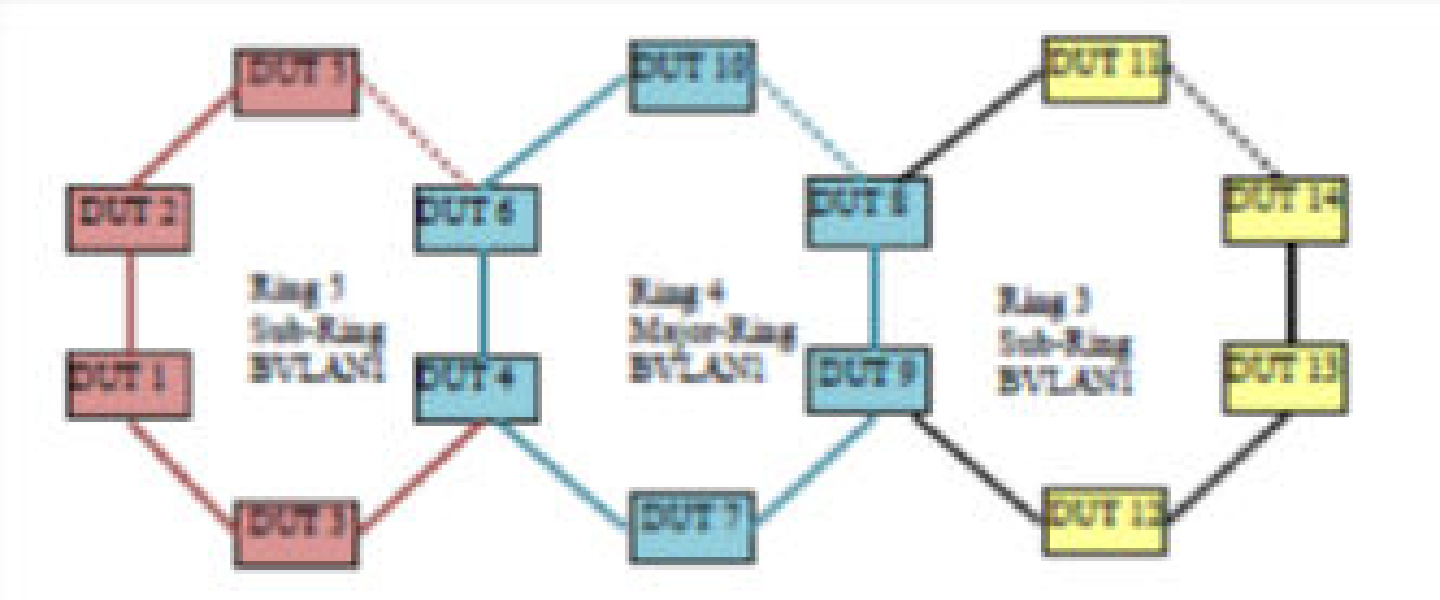
The interconnection nodes play a vital role in traffic forwarding and should be aware of ERP instances associated with each other within the node. In ERP over ESI VLAN (Brocade NetIron CES Series and Brocade NetIron CER Series) on page 513, the DUT17 interconnection node needs to be aware of Ring 1 being a sub-ring of the Ring 2 ERP instance. This association is important as it is required to perform a FDB flush in Ring-2 on receiving R-APS (SF) messages from Ring-1 .

Interconnection rings with same VLANs

The PBB case such as in Figure 146, where one B-VLAN span across multiple rings, the ring mapping will still be maintained as it helps push the events from sub-rings to major ring.

The PB case will be also handled similarly.

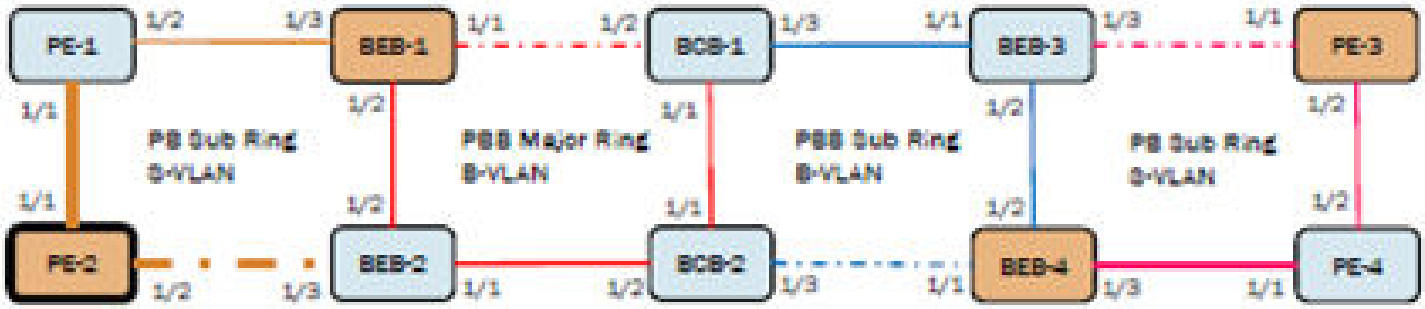
FIGURE 143 ERP configuration with same VLAN



Sample configurations

PB ring node

FIGURE 144 PB ring node sample configuration



PE-2 Configuration with S-VLAN 200:

```
!
esi svlan encapsulation svlan
vlan 200
tagged ethe 1/1 ethe 1/2
!
erp 200
left-interface esi svlan vlan 200 ethe 1/1
right-interface esi svlan vlan 200 ethe 1/2
rpl-owner
sub-ring
rpl esi svlan vlan 200 ethe 1/2
enable
!
interface ethernet 1/1
port-type provider-network
enable
!
interface ethernet 1/2
port-type provider-network
enable
```

PBB interconection node (BEB)

FIGURE 145 PBB interconection node (BEB) sample configuration

BEB-1 Configuration for Major ring B-VLAN 100

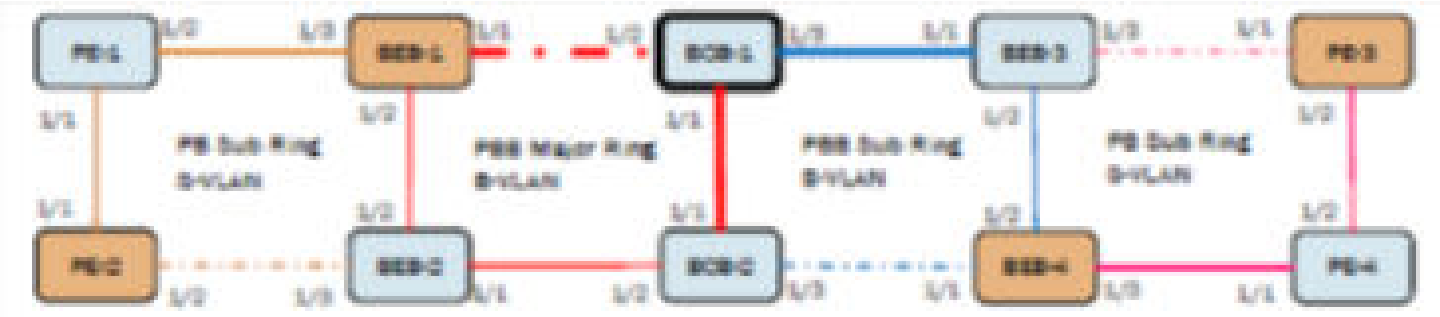
```
!
esi bvlan encapsulation bvlan
vlan 100
tagged ethe 1/1 ethe 1/2
!
erp 100
left-interface esi bvlan vlan 100 ethe 1/1
right-interface esi bvlan vlan 100 ethe 1/2
rpl-owner
raps-default-mac
rpl esi bvlan vlan 100 ethe 1/1
enable
!
interface ethernet 1/1
port-type backbone-network
enable
!
interface ethernet 1/2
port-type backbone-network
enable
```

BEB-1 Configuration for Sub-ring S-VLAN 200

```
!
esi svlan encapsulation svlan
vlan 200
tagged ethe 1/3
!
erp 200
right-interface esi svlan vlan 200 ethe 1/3
raps-default-mac
sub-ring parent-ring-id 100
enable
!
interface ethernet 1/3
port-type backbone-edge
enable
!
```

PBB interconection node (BCB)

FIGURE 146 PBB interconnection node (BCB)



BCB-1 Configuration for Major ring B-VLAN 100

```
!
esi bvlan encapsulation bvlan
vlan 100
tagged ethe 1/1 ethe 1/2
```

```
!
erp 100
left-interface esi bvlan vlan 100 ethe 1/1
right-interface esi bvlan vlan 100 ethe 1/2
raps-default-mac
rpl esi bvlan vlan 100 ethe 1/2
enable
!
interface ethernet 1/1
port-type backbone-network
enable
!
interface ethernet 1/2
port-type backbone-network
enable
```

ERP support for PBB (Brocade NetIron MLX Series and Brocade NetIron XMR Series)

To support ERP protocol over a PBB network on the Brocade NetIron MLX Series and Brocade NetIron XMR Series platforms will make use of topology groups. The ERP protocol will be run on a regular L2 VLAN which will be the master VLAN of the topology group and the VPLS VLANs which will carry the PB/PBB traffic will be member VLANs of the topology group.

Configuration requirements

- Configure regular L2 VLANs for ERP operation
- Configure PBB VPLS VLANs for carrying PB/PBB traffic
- Configure the topology group
  - ERP protocol over the master VLAN
  - PB/PBB traffic over the member VPLS VLANs

Blocking of L2 protocols for PBB

The configurations discussed will be blocked

- ERP is the only protocol that will be supported for PBB, any other protocol configuration with PBB is blocked.
- If a topology group is configured with PBB VPLS member VLANs, then only ERP can be enabled on the master VLAN.
- If the master VLAN of a topology group is enabled with a protocol other than ERP, then PBB VPLS member VLAN configuration will not be allowed on that topology group.

Sample configurations

PB Ring node

Figure to be added here for PB ring node

PE-2 ERP Configuration with regular VLAN 201

```
vlan 201
tagged ethe 1/1 ethe 1/2
!
erp 201
left-interface vlan 201 ethe 1/1
right-interface vlan 201 ethe 1/2
```

```
    sub-ring
    raps-default-mac
    rpl vlan 201 ethe 1/2
    enable
    !
```

PE-2 Topology group configuration

```
    !
    topology-group 1
    master-vlan 201
    member-vlan vpls id 1 vlan 200

!
```

PE-2 PBB Configuration with S-VLAN 200

```
    !
    tag-type 88a8 eth 1/1
    tag-type 88a8 eth 1/2
    !
    router mpls
        vpls pb-pbb 1
            pbb
            vlan 200
            tagged ethe 1/1 ethe 1/2
```

Figure to be added here for PE-2 PBB Configuration with S-VLAN

BEB-1 ERP PB sub-ring with regular VLAN 201

```
    !
    vlan 201
    tagged ethe 1/3
    !
    erp 201
    right-interface vlan 201 ethe 1/3
    raps-default-mac
    sub-ring parent-ring-id 100
    enable
    !
```

BEB-1 topology group for PB sub-ring

```
    !
    topology-group 1
    master-vlan 201
    member-vlan vpls id 1 vlan 200
    !
```

BEB-1 PB configuration with S-VLAN 200 and PBB configuration with B-VLAN 100 and ISID 10000

```
    !
    vlan 100
    tagged eth 1/1 eth 1/2
    !
    tag-type 88a8 eth 1/1
    tag-type 88a8 eth 1/2
    tag-type 88a8 eth 1/3
    !
    router mpls
    vpls pb-pbb 1
    pbb
    vlan 200
    tagged ethe 1/3
    vlan 100 isid 10000
```

```
tagged ethe 1/1 to 1/2
!
```

PBB Interconnection node (BCB)

Figure to be added here for PBB Interconnection node (BCB)

BCB-1 Configuration for Major ring B-VLAN 100

```
!
vlan 100
tagged ethe 1/1 ethe 1/2
!
tag-type 88a8 eth 1/1
tag-type 88a8 eth 1/2
!
erp 100
left-interface vlan 100 ethe 1/1
right-interface vlan 100 ethe 1/2
raps-default-mac
rpl vlan 100 ethe 1/2
enable
!
```

BCB-1 Configuration for Sub ring B-VLAN 100

```
!
vlan 100
tagged ethe 1/3
!
tag-type 88a8 eth 1/3
!
erp 101
left-interface vlan 100 ethe 1/3
sub-ring parent-ring-id 100
raps-default-mac
enable
!
```

Parent ring ID configuration

Figure to be added here for Parent Ring ID configuration

- Each sub-ring will be connected to either a major ring or a sub-ring which is referred to as a parent ring.
- In the figure above, the ERP instance 1 is the parent ring for sub-ring ERP instance 2 and ERP instance 2 is the parent ring for sub-ring ERP instance 3.
- If both the major ring and sub-ring are running on the same VLAN then the parent ring can be identified based on the VLAN, However, if they are running on different VLANs (especially in PB/PBB networks) with multiple ERP instances then the administrator must specify the parent ring ID using the **sub-ring parent-ring-id** command.
- The sub-ring parent-ring-id must be configured on the sub-ring ERP instance of an interconnection node.

RAPS-propagate-tc

Figure to be added here for RAPS-propagate-tc

- The **RAPS-propagate-tc** command must be configured on the sub-ring ERP instance.
- When the **raps-propagate-tc** command is configured, any topology change on the sub-ring will be communicated to the major ring by sending a R-APS(Flush event). This results in a FDB flush on all the major ring nodes.
- The RAPS-propagate-tc configuration is required only when a node is present in between two interconnection nodes (for example: Node-4 in above topology).

Node-3 Configuration for Major ring VLAN 100

```
!
vlan 100
tagged ethe 1/1 ethe 1/3
!
erp 100
left-interface vlan 100 ethe 1/3
right-interface vlan 100 ethe 1/1
raps-default-mac
enable
!
```

Node-3 Configuration for Sub ring VLAN 100

```
!
vlan 100
tagged ethe 1/2
!
erp 200
right-interface vlan 100 ethe 1/2
raps-default-mac
sub-ring parent-ring-id 100
raps-propagate-tc
rpl vlan 100 ethe 1/2
enable
!
```

Figure to be added here for RAPS-propagate-tc configuration

A and B are end-points exchanging traffic, active path is as indicated by green line.

Figure to be added here for RAPS-propagate-tc topology change

- When the link connecting Node-1 and Node-2 goes down, it results in topology change on the sub-ring.
- To restore the traffic between endpoints A and B, an FDB flush is performed on all sub-ring nodes, interconnection nodes (Node-3 and Node-5) and Node-4.

The sequence of events for restoring the traffic

When the link connecting Node-1 and Node-2 goes down, it results in topology change on sub-ring resulting in following events:

- The topology change on the sub-ring results in the generation of R-APS(SF) PDUs by Node-1 and Node-2 in addition to flushing their own FDBs.
- Node-3 and Node-5 on the reception of R-APS(SF) perform an FDB flush on the sub-ring ERP interface 1/2.
- Node-3 and Node-5 will also perform an FDB flush on the major ring interfaces 1/1 and 1/3. The major ring on which the FDB needs to be flushed is identified by the configured parent-ring-id.
- If the **raps-propagate-tc** command is configured on the sub-ring instance of Node-3 and Node-5, it will result in the generation of R-APS(Flush event) PDUs on major ring interfaces, on reception of this PDU all major ring nodes will perform FDB flush.

After the above events, the active path changes as indicated by green line in the figure above.

In the figure above, only Node-4 needs to be flushed to restore traffic between endpoints A and B. However, other nodes (Node-6 and Node-7) will also flush due to reception of R-APS(Flush event) as per standard. Due to this, raps-propagate-tc must be configured only when a node (Node-4) exists between interconnection nodes (Node-3 and Node-5).

Viewing ERP operational status and clearing ERP statistics

You can view operational status and statistics and clear statistics for all links or particular links.

Viewing ERP operational status and statistics

To view ERP statistics, enter the following command on the RPL owner:

Syntax: show erp [*enter* | *erp_id*]

To view ERP information for all links, enter **show erp** command followed by pressing the Enter key (carriage return). To view statistics for a particular link, enter the ERP ID after the command.

Example output:

```
device #show erp 7
ERP 7(version 2)- VLAN 504
==========================================================================
Erp ID Status Oper Node Topo
state role group
1 enabled Idle rpl-owner -
Ring type WTR WTB Guard Holdoff Msg
time(min) time(ms) time(ms) time(ms) intv(ms)
Major-ring 5 7000 2000 0 1000
I/F Port ERP port state Interface status Interface type
L 1/12 blocking normal rpl
R 1/11 forwarding normal non-rpl
RAPS sent RAPS rcvd RAPS dropped RAPS ignored Oper state changes
3 3 0 0 0
```

Table 64 summarizes the table fields and their meanings.

TABLE 64 Summary of CLI output for show erp command

This field...	Displays...
ERP id	The ERP ID is the number that was configured at setup. The ERN appends this number to the permanent portion of the MAC address (01-19-A7-00-00) used for ERP.
Status	Enabled or disabled
Operational state	Init, Idle, Protection, Manual Switch, Forced Switch or Pending
Node role	rpl-owner, non-rpl-node or rpl-node
Topology group	<topology group id> or "- " (- means N/A)
Ring type	Major-ring or Sub-ring
Timers	Configuration value for each timer
Interfaces (I/F)	L (left) or R (right)
Port	<slot/port>
ERP port state:	disabled, blocking, forwarding
Interface status:	normal, signal-fail, manual-switch or forced-switch
Interface type:	rpl or non-rpl
RAPS sent:	RAPS sent by MP (self generated)
RAPS rcvd:	RAPS received by MP

TABLE 64 Summary of CLI output for show erp command (continued)

This field...	Displays...
RAPS dropped:	RAPS dropped by MP
RAPS ignored:	RAPS ignored (for example, the guard-timer, or non regular type)

Clearing ERP statistics

You can clear ERP statistics by entering the **clear erp statistics** command and the specific *erp_id* to clear the statistics of one erp instance. You can clear all ERP statistics by entering the **clear erp statistics** command to clear the statistics of all erp instances.

```
device#
clear erp
7 statistics
```

Syntax: clear erp *erp_id* **statistics**

Virtual Switch Redundancy Protocol (VSRP)

Virtual Switch Redundancy Protocol

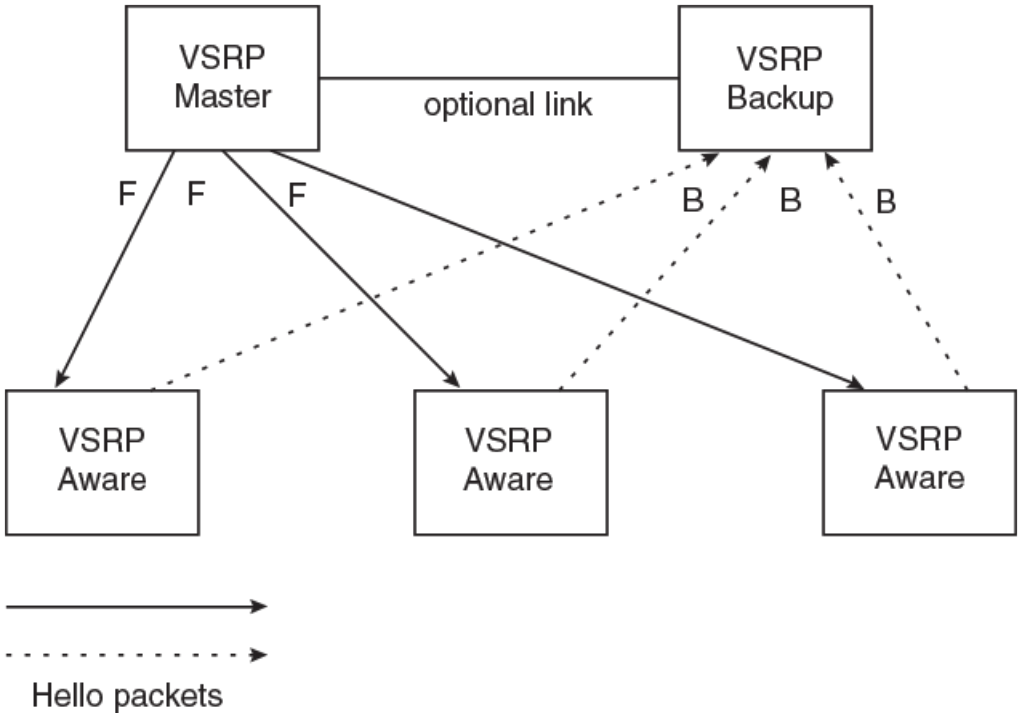
Virtual Switch Redundancy Protocol (VSRP) is a proprietary protocol that provides redundancy and sub-second failover in Layer 2 mesh topologies. Based on the Virtual Router Redundancy Protocol Extended (VRRP-E), VSRP provides one or more backups for the Brocade device. If the active Brocade device becomes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

VSRP is a Brocade proprietary protocol that provides redundancy and sub-second failover in Layer 2 and Layer 3 mesh topologies. Based on the Brocade's proprietary VRRP-E, VSRP provides one or more backups for the device. If the active device becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

Layer 2 and Layer 3 share the same VSRP configuration information.

Figure 150 shows a VSRP configuration.

FIGURE 147 VSRP mesh - redundant paths for Layer 2 traffic



In this example, two Brocade devices are configured as redundant paths for VRID 1. On each Brocade device, a Virtual Router ID (VRID) is configured on a port-based VLAN. Since VSRP is primarily a Layer 2 redundancy protocol, the VRID applies to the entire VLAN. However, you can selectively remove individual ports from the VRID if needed.

Following Master election (described below), one of the devices becomes the Master for the VRID and sets the state of all the VLAN's ports to Forwarding. The other device is a Backup and sets all the ports in its VRID VLAN to Blocking.

If a failover occurs, the Backup becomes the new Master and changes all its VRID ports to the Forwarding state.

Other devices can use the redundant paths provided by the VSRP devices. In this example, three devices use the redundant paths. A device that is not itself configured for VSRP but is connected to a device that is configured for VSRP, is VSRP aware . In this example, the three devices connected to the VSRP devices are VSRP aware. A device that is VSRP aware can failover its link to the new Master in sub-second time, by changing the MAC address associated with the redundant path.

When you configure VSRP, make sure each of the non-VSRP devices connected to the VSRP devices has a separate link to each of the VSRP devices.

NOTE

If the Brocade device is configured as the VSRP Master and it is connected to a FastIron switch (FESX, FSX, SuperX, FGS, and FLS) that is operating as a VSRP-Aware device, the FastIron switch must have the **vsrp-aware tc-vlan-flush** command configured at the VLAN level.When the **vsrp-aware tc-vlan-flush** command is enabled on the FastIron switch, MAC addresses will be flushed at the VLAN level, instead of at the port level. MAC addresses will be flushed for every topology change (TC) received on the VSRP-aware ports.

Layer 2 redundancy

VSRP provides Layer 2 redundancy. This means that Layer 2 links are backed up.

You can configure VSRP to provide redundancy for Layer 2 only or also for Layer 3:

- Layer 2 only - The Layer 2 links are backup up but specific IP addresses are not backed up.
- Layer 2 and Layer 3 - The Layer 2 links are backup up and a specific IP address is also backed up. Layer 3 VSRP is the same as VRRP-E. However, using VSRP provides redundancy at both layers at the same time.

The Brocade device supports Layer 2 and Layer 3 redundancy. You can configure a Brocade device for either Layer 2 only or Layer 2 and Layer 3. To configure for Layer 3, specify the IP address you are backing up.

NOTE
If you want to provide Layer 3 redundancy only, disable VSRP and use VRRP-E.

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:

- Brocade **devices** - The Brocade device 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:

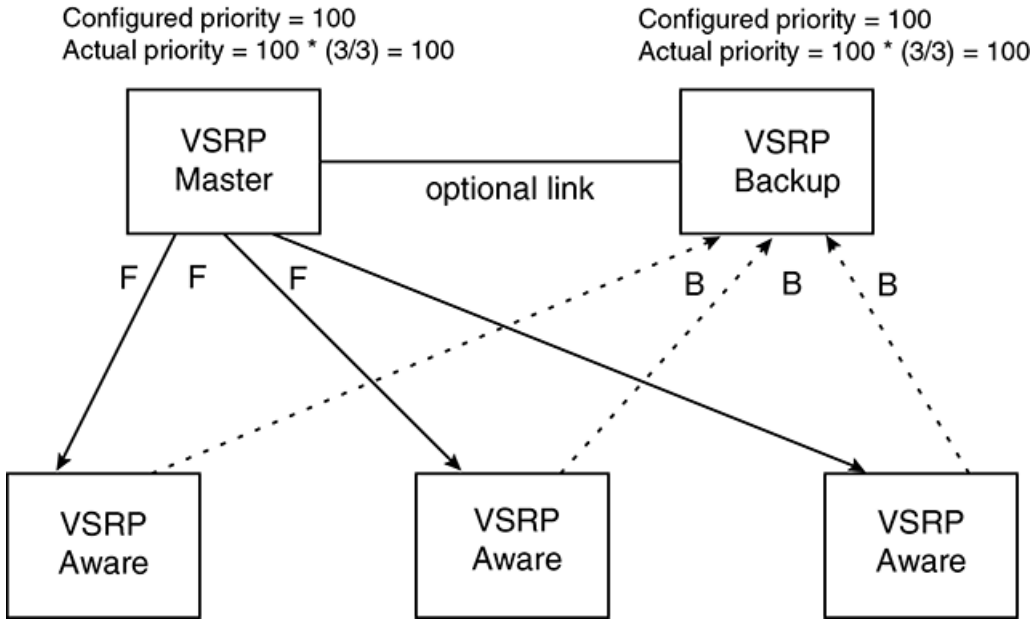
- If the Backup receives a Hello message with a higher priority than its own, the Backup resets its Dead Interval and returns to normal Backup status.
- If the Backup does not receive a Hello message with a higher priority than its own by the time the hold-down timer expires, the Backup becomes the new 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

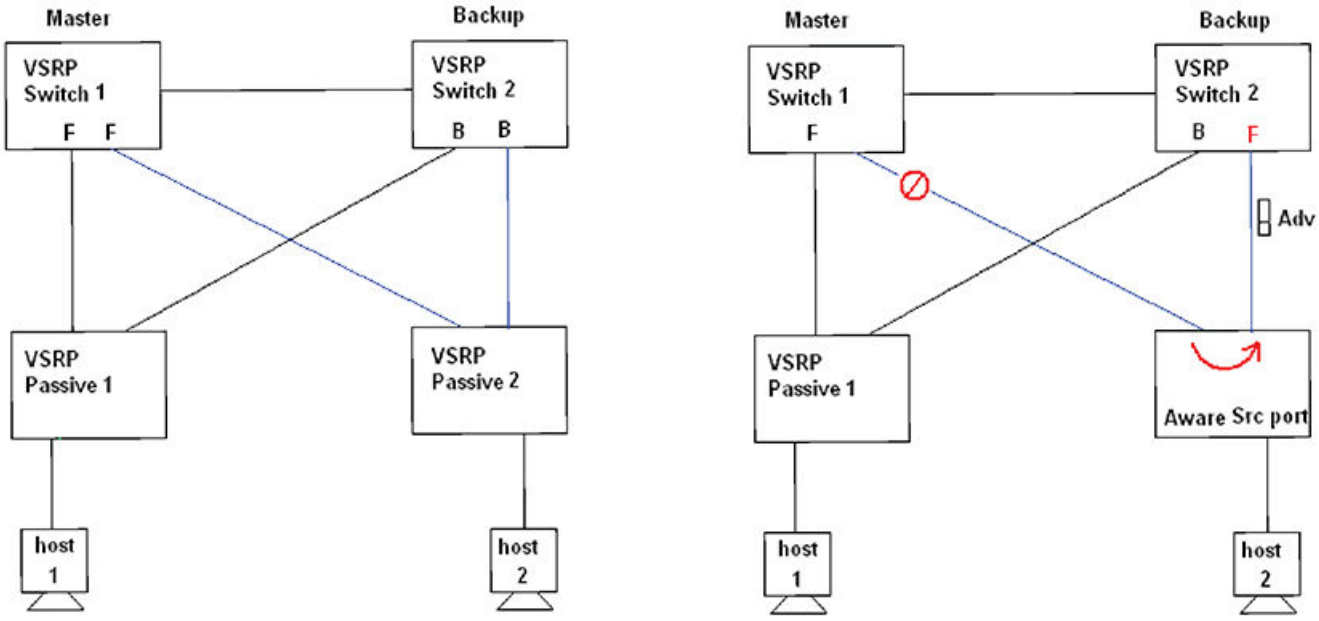
Each VSRP device has a VSRP priority for each VRID and its VLAN. The VRID is used during Master election for the VRID. By default, a device's VSRP priority is the value configured on the device (which is 100 by default). However, to ensure that a Backup with a high number of up ports for a given VRID 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 151.

FIGURE 148 VSRP priority



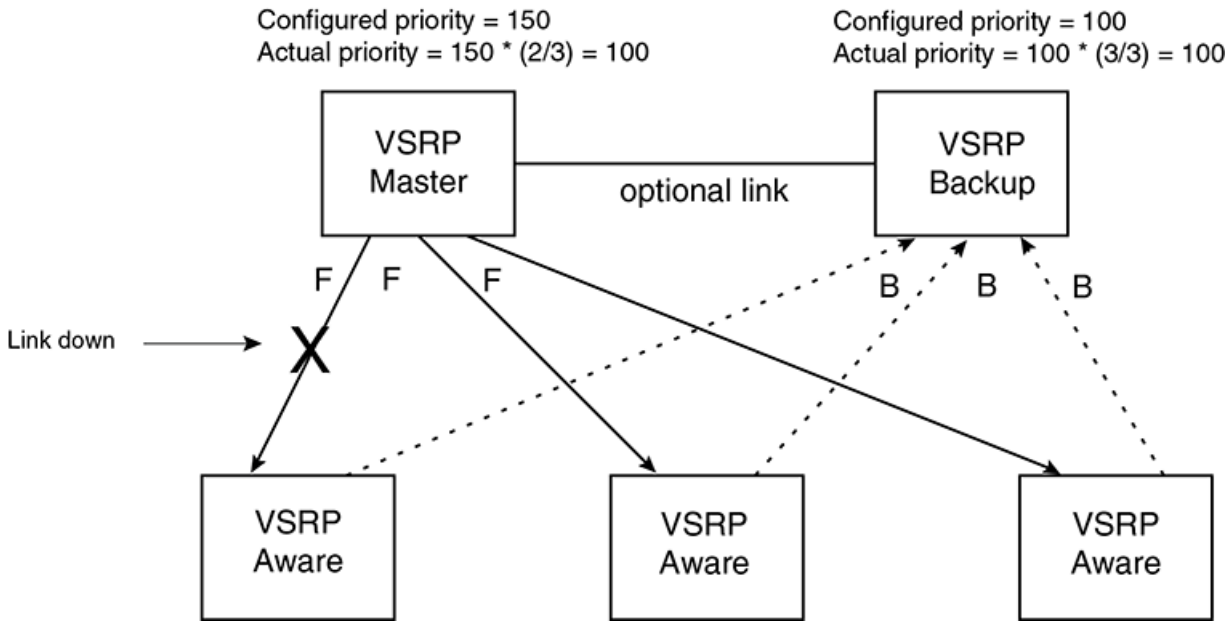
However, if one of the VRID's ports goes down on one of the Backups, that Backup's priority is reduced. If the Master's priority is reduced enough to make the priority lower than a Backup's priority, the VRID fails over to the Backup. VSRP priority calculation shows an example.

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 VSRP priority calculation 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 152.

FIGURE 149 VSRP priority bias

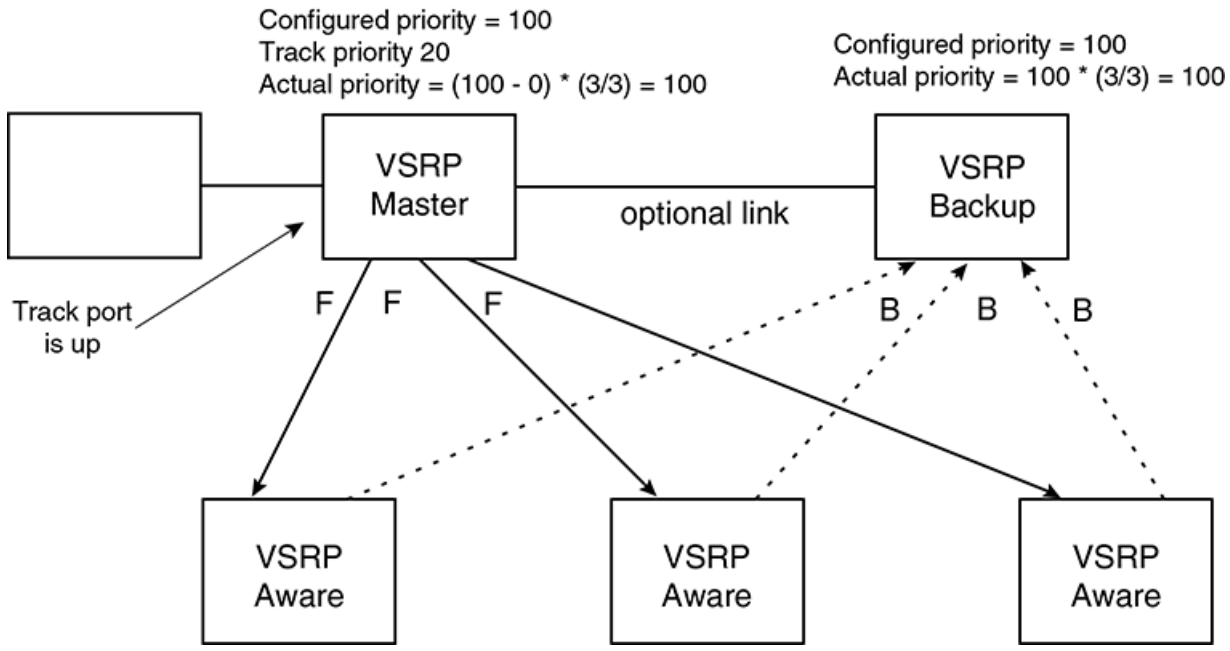


Track ports

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.

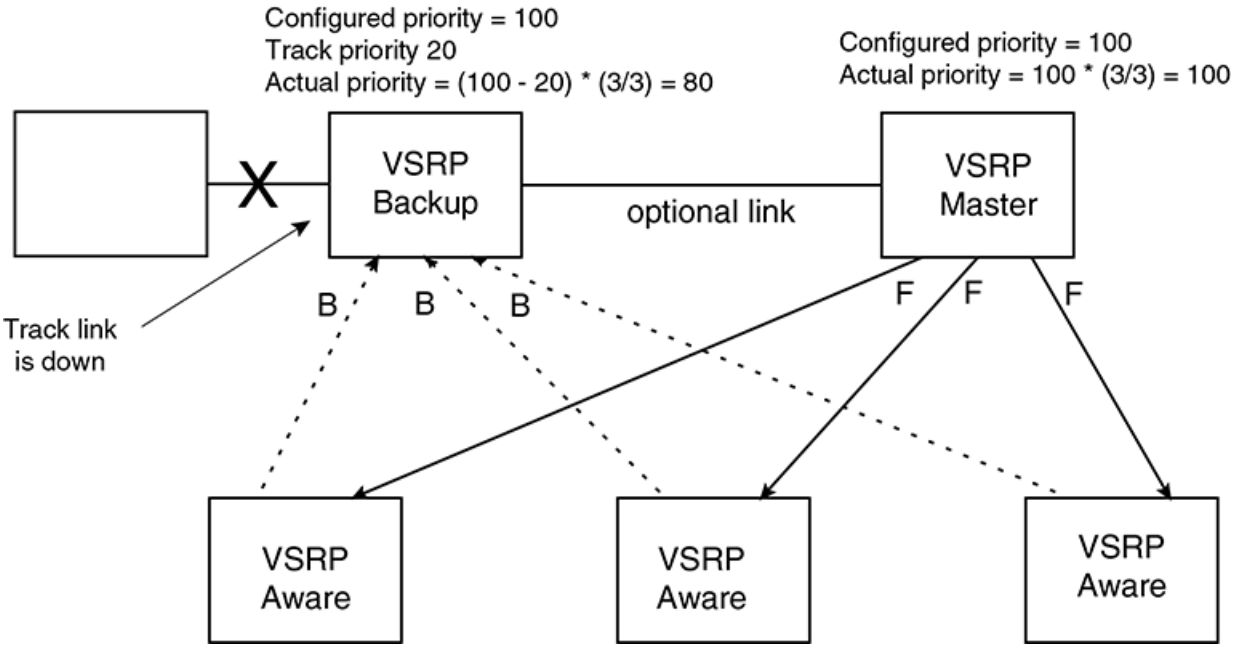
When you configure a track port, you assign a priority value to the port. If the port goes down, VSRP subtracts the track port's priority value from the configured VSRP priority. For example, if the you configure a track port with priority 20 and the configured VSRP priority is 100, the software subtracts 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 153 shows an example.

FIGURE 150 Track port priority



In Figure 153, 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 154.

FIGURE 151 Track port priority subtracted during priority calculation



MAC address failover on VSRP-aware devices

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

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

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

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

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

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

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

Configuring basic VSRP parameters

To configure VSRP, perform the following required tasks:

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

  NOTE
  If you already have a port-based VLAN but only want to use VSRP on a sub-set of the VLANs ports, you can selectively remove ports from VSRP service in the VLAN. Refer to Removing a port from the VRID's VLAN on page 537.

- Configure a VRID:
  - Specify that the device is a backup. Since VSRP, like VRRP-E, 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.

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

Syntax: [no] vsrp vrid num

The *num* parameter specifies the VRID and can be from 1 - 255.

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

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

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

- Changing the backup priority on page 537

- Changing the default track priority on page 540

Syntax: [no] enable disable |

Note on VSRP support when using ESI

VSRP is supported only for VLANs that are part of the default ESI. VSRP is not supported for VLANs configured under user-defined ESIs.

Configuring optional VSRP parameters

The following sections describe how to configure optional VSRP parameters.

Enabling Layer 3 VSRP

Layer 2 VSRP is enabled globally by default on the device; it just needs to be activated or enabled on a VRID. If you want to use Layer 3 VSRP, you must enable it by entering the following command at the CONFIG level.

```
device(config)# router vsrp
```

Syntax: [no] ID="router router vsrp

If you want to provide Layer 3 redundancy only, you could use VRRP or VRRP-Extended. You may use **router vrrp** or **router vrrp-extended** commands as long as **router vsrp** command is not enabled.

Disabling or re-enabling VSRP

To disable Layer 3 VSRP, enter the following command at the global CONFIG level.

```
device(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.

```
device(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.

```
Brocade(config-vlan-200)# vsrp auth-type simple-text-auth ourpword
```

This command configures the simple text password "ourpword".

Syntax: [no] 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 8 character password. If you use this parameter, make sure all interfaces on all the devices supporting this VRID are configured for simple password authentication and use the same password.

VSRP 2

In VSRP setup, there are always at least two VSRP switches for each VSRP instance. A passive device should always have either one access link or one trunk link connected with each VSRP switch for each VSRP instance. This can create a black hole scenario. A black hole is when VSRP failover causes data traffic from the switches/hosts which connect to VSRP passive switch to go nowhere.

VSRP 2 can detect the health of each pair/set of links for each VSRP instance.VSRP 2 detects which link of VSRP backup switch not receiving any advertisement for specific time duration. The VSRP backup switch can treat the link of the pair on the master side is down or broken and set its own link of the pair in forwarding state. The VSRP backup switch sends out gratuitous ARP for VSRP master only to this link. Other links of other VSRP instances in VSRP backup switch are still in blocking state as shown in figure Figure 155,Figure 156, and Figure 157.

FIGURE 152 Black hole scenario 1

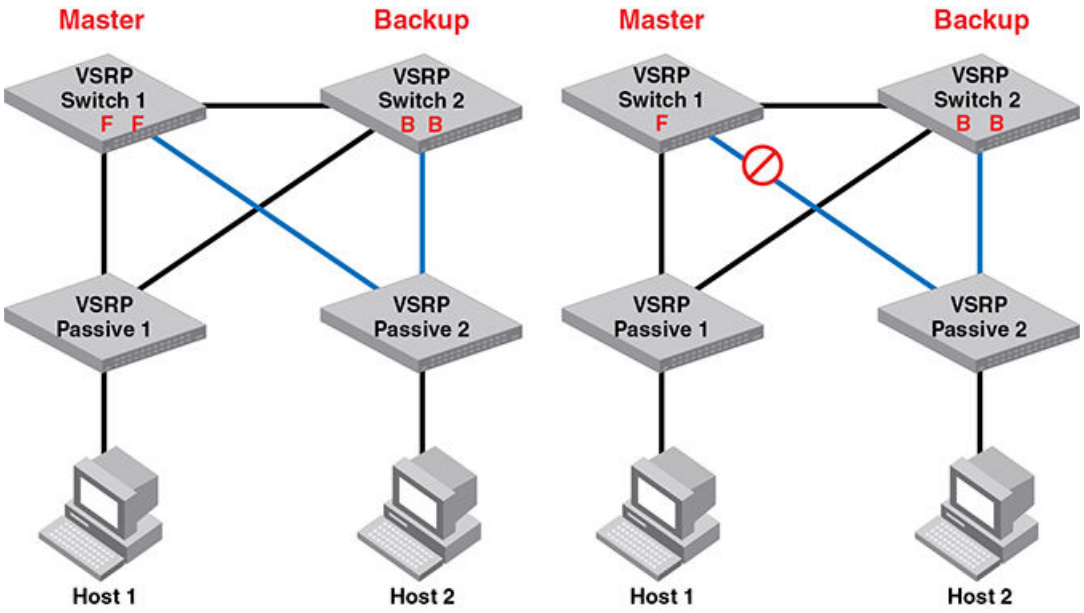


FIGURE 153 Black hole scenario 2

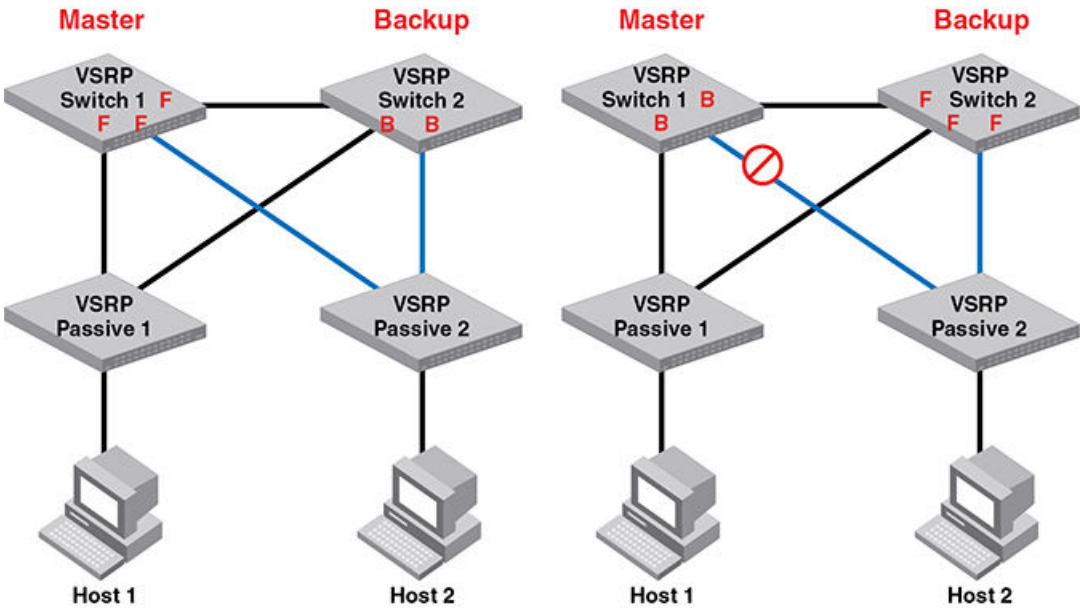
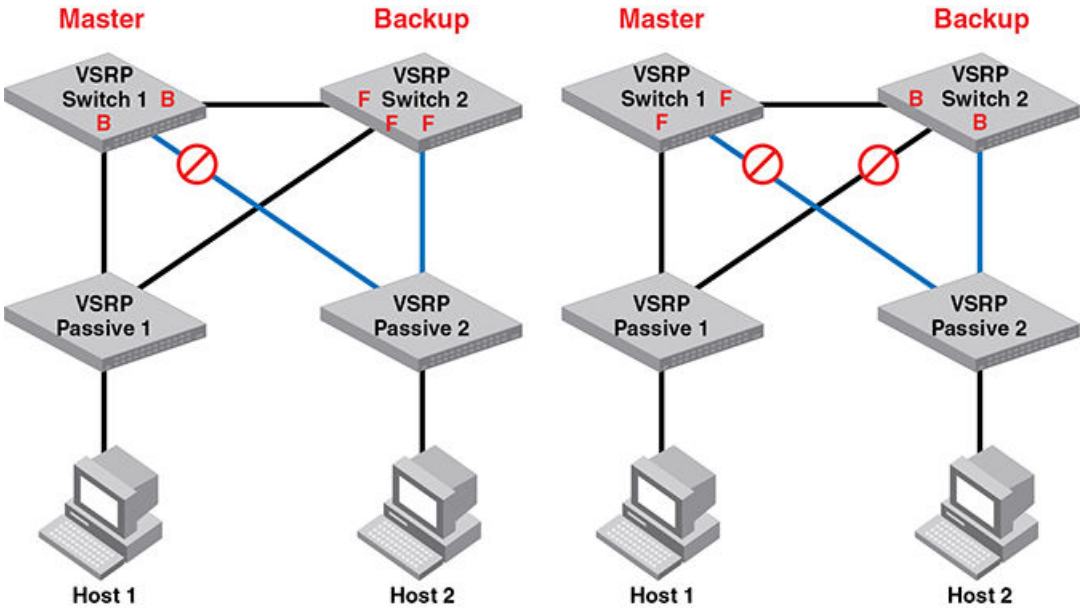


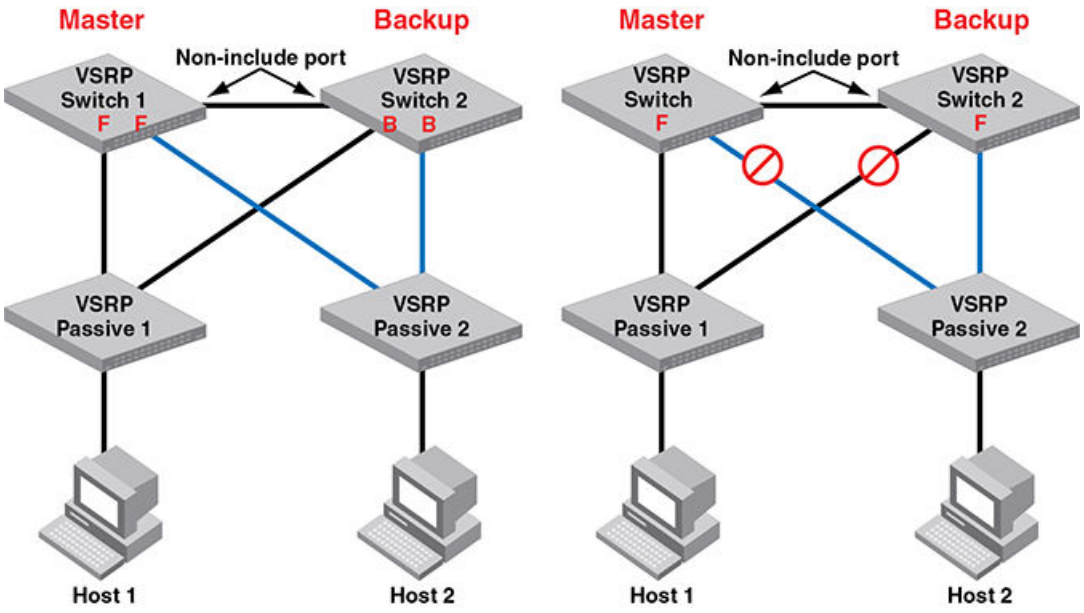
FIGURE 154 Black hole scenario 3



VSRP failover:

- VSRP backup set all include links in blocking state. Blocking ports drop data traffic.
- VSRP failover changes master state by current priority change.
- Current priority changes by link failure and track port failure.

FIGURE 155 Correct VSRP behavior



VSRP is switch redundancy, VSRP 2 is link redundancy.

When VSRP backup changes an include port from blocking state to forwarding state, to make the aware session changes, VSRP backup will send advertisements on the forwarding include ports every 3*hello time.

VSRP aware switches are able to change the src port of aware session and flush MACs.

For VSRP non-aware switches (other vendors), the non-aware switch will flush MACs because the link connecting VSRP master is failed.

VSRP backup will toggle the interface when it sets an include port to forwarding state by VSRP 2.

VSRP 2 doesn't change the master/backup state, only changes the port state.

The change of Master/backup state (VSRP failover) still follows the rules of current priority of VSRP.

VSRP 2 supports:

- hold-down time
- track port
- preempt-mode
- restart port
- topology groups
- VLAN groups.
- VPLS VLAN by topology group.
- Layer 3 VSRP with a condition: non-include link in between two VSRP routers is a must.

Configuration considerations:

- If multiple VSRP instances on multiple VLAN are configured the on one side of link pair, the same VSRP instances of same VLANs must configure on another side of link pair.

- Link-pair has to be enabled or disabled on all VSRP switches for the same VSRP instance.
- Currently, VSRP 2 only supports two VSRP switches in the topology. Multiple VSRP switches may cause a loop when the link redundancy set the VSRP ports in the forwarding state in the link failed cases.
- For VSRP 2 supporting Layer 3 VSRP, it is necessary to have a non-include link in between two VSRP switches. VSRP virtual router is still in VSRP master. Layer 3 data traffic is switched by VSRP backup to VSRP master. The traffic is routed by VSRP master (virtual router).

Configuring VSRP 2

Configure VSRP seeConfiguring basic VSRP parameters on page 530, then use the **link-redundancy** command at the interface level to enable link redundancy. To enable link redundancy, use commands such as the following.

```
device(config)# vlan 10
device(config-vlan-10)# vsrp vrid 10
device(config-vlan-10-vsrp-10)# link-redundancy
```

After enabling link redundancy, VSRP switches to master-confirm state and the backup state creates a link redundant port list.

VSRP switches that are in initial state and master state don't need to create link redundant port list

Syntax: [no] vsrp-linkpair-id *id*

Displaying VSRP 2

To display VSRP, use the **show vsrp** command as shown below.

```
device#show vsrp
VLAN 10
 Auth-type no authentication
 VRID 1
 ========
  State           Administrative-status Advertise-backup Preempt-mode Link-Red
  Backup          Enabled               Enabled          True         Enabled
  Parameter        Configured Current    Unit/Formula
  Priority         100        100                (100-0)*(3.0/3.0)
  Hello-interval   1          1                   sec/1
  Dead-interval    3          3                   sec/1
  Hold-interval    3          3                sec/1
  Initial-ttl      2          2                hops
  Backup-Hello     60                                   60                   sec/1
  Master router 10.243.150.0 or MAC xxxx.dbf3.9600 expires in 00:00:03
  Member ports:      ethe 2/14 ethe 2/18 to 2/19
  Operational ports: ethe 2/14 ethe 2/18 to 2/19
   Forwarding ports:  ethe 2/14
Link-Redundancy-port:
  port 2/14 status FORWARD
  port 2/18 status BLOCK
  port 2/19 status BLOCK
```

Syntax: show vsrp [**vrid** *num* | **vlan** *vlan-id*]

This display shows the following information when you use the **vrid***num* or **vlan***vlan-id* parameter. For information about the display when you use the **aware** parameter, refer to Displaying the active interfaces for a VRID on page 544.

TABLE 65 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.

TABLE 65 CLI display of VSRP VRID or VLAN information (continued)

This field...	Displays...
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	This device's VSRP state for the VRID. The state can be one of the following: • initialize - VSRP is not enabled on the VRID. If the state remains "initialize" after you enable VSRP on the VRID, make sure that the VRID is also configured on the other routers Routing Switches and that the routers Routing Switches can communicate with each other. NOTE If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the VRID. • standby - This device is a Backup for the VRID. • master - This device is the Master for the VRID.
Administrative-status	The administrative status of the VRID. The administrative status can be one of the following: • disabled - The VRID is configured on the interface but VSRP or VRRP-E has not been activated on the interface. • enabled - VSRP has been activated on the interface.
Advertise-backup	Whether the device is enabled to send VSRP Hello messages when it is a Backup. This field can have one of the following values: • disabled - The device does not send Hello messages when it is a Backup. • enabled - The device does send Hello messages when it is a Backup.
Preempt-mode	Whether the device can be pre-empted by a device with a higher VSRP priority after this device becomes the Master. This field can have one of the following values: • disabled - The device cannot be pre-empted. • enabled - The device can be pre-empted.
NOTE For the following fields: • Configured - indicates the parameter value configured on this device. • Current - indicates the parameter value received from the Master. • Unit - indicates the formula used tor calculating the VSRP priority and the timer scales in effect for the VSRP timers. A timer's true value is the value listed in the Configured or Current field divided by the scale value.	
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.

TABLE 65 CLI display of VSRP VRID or VLAN information (continued)

This field...	Displays...
	If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. **NOTE** If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. **NOTE** A Foundry MRP ring counts as one hop, regardless of the number of nodes in the ring.
next hello sent in time or next backup hello sent in time	The amount of time until the Master/Backup sends its next hello message.
master router or backup router	The IP address or MAC address of the master or backup router, and the amount of time remaining until the dead interval expires.
Member ports	The ports in the VRID.
Operational ports	The member ports that are currently up.
Forwarding ports	The member ports that are currently in the Forwarding state. Ports that are forwarding on the Master are listed. Ports on the Standby, which are in the Blocking state, are not listed.
Link-Redundancy-port	The status os the port on which link redundancy has been configured.

Removing a port from the VRID's VLAN

By default, all the ports in the VLAN on which you configure a VRID are interfaces for the VRID. You can remove a port from the VRID while allowing it to remain in the VLAN.

Removing a port is useful in the following cases:

- There is no risk of a loop occurring, such as when the port is attached directly to an end host.
- You plan to use a port in a Foundry MRP ring.

To remove a port from a VRID, enter a command such as the following at the configuration level for the VRID.

```
device(config-vlan-200-vrid-1)# no include-port ethernet 1/2
```

Syntax: [no] include-port ethernet slot/portnum

The **ethernet** *slot/portnum* parameter specifies the port you are removing from the VRID. The port remains in the VLAN but its forwarding state is not controlled by VSRP.

Changing the backup priority

When you enter the backup command to configure the device as a VSRP Backup for the VRID, you also can change the backup priority and the track priority:

- The backup priority is used for election of the Master. The VSRP Backup with the highest priority value for the VRID is elected as the Master for that VRID. The default priority is 100. If two or more Backups are tied with the highest priority, the Backup with the highest IP address becomes the Master for the VRID.
- The track priority is used with the track port feature. Refer to VSRP priority calculation on page 525 and Changing the default track priority on page 540.

To change the backup priority, enter a command such as the following at the configuration level for the VRID.

```
device(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 8- 255. The default is 100.

For a description of the **track-priority** *value* parameter, refer to Changing the default track priority on page 540.

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.

```
device(config-vlan-200-vsrp-1)# save-current-values
```

Syntax: [no] save-current-values

Changing the Time-To-Live (TTL)

A VSRP Hello packet's TTL specifies how many hops the packet can traverse before being dropped. A hop can be a router or a Layer 2 Switch. You can specify from 1 - 255. The default TTL is 2. When a VSRP device (Master or Backup) sends a VSRP HEllo packet, the device subtracts one from the TTL. Thus, if the TTL is 2, the device that originates the Hello packet sends it out with a TTL of 1. Each subsequent device that receives the packet also subtracts one from the packet's TTL. When the packet has a TTL of 1, the receiving device subtracts 1 and then drops the packet because the TTL is zero.

> NOTE
> A Foundry MRP ring is considered to be a single hop, regardless of the number of nodes in the ring.

To change the TTL for a VRID, enter a command such as the following at the configuration level for the VRID.

```
device(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.

```
device(config-vlan-200-vrid-1)# hello-interval 10
```

Syntax: [no] hello-interval num

The *num* parameter specifies the interval and can be from 1 - 28 units of 100 milliseconds. The default is 1 unit of 100 ms.

NOTE
The default Dead interval is three times the Hello interval plus one-half second. Generally, if you change the Hello interval, you also should change the Dead interval on the Backups.

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the Dead interval

The Dead interval is the number of seconds a Backup waits for a Hello message from the Master before determining that the Master is dead.

To change the Dead interval, enter a command such as the following at the configuration level for the VRID.

```
device(config-vlan-200-vrid-1)# dead-interval 3
```

Syntax: [no] dead-interval num

The *num* parameter specifies the interval and can be from 3 - 84 units of 100 milliseconds. The default is 3 units of 100 ms (300 milliseconds).

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the Backup Hello state and interval

By default, Backups do not send Hello messages to advertise themselves to the Master. You can enable these messages if desired and also change the message interval.

To enable a Backup to send Hello messages to the Master, enter a command such as the following at the configuration level for the VRID.

```
device(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. The interval be from 600 - 3600 units of 100 milliseconds. The default is 600 units of 100 milliseconds.

To change the Backup Hello interval, enter a command such as the following at the configuration level for the VRID.

```
device(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 600 - 3600 units of 100 milliseconds. The default is 600 units of 100 milliseconds.

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.

```
device(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 2 - 84 units of 100 milliseconds. The default is 2 units of 100 ms (200 milliseconds).

NOTE
If you change the timer scale, the change affects the actual number of seconds.

Changing the default track priority

When you configure a VRID to track the link state of other interfaces, if one of the tracked interface goes down, the software changes the VSRP priority of the VRID interface.

The software reduces the VRID priority by the amount of the priority of the tracked interface that went down. For example, if the VSRP interface's priority is 100 and a tracked interface with track priority 60 goes down, the software changes the VSRP interface's priority to 40. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

The default track priority for all track ports is 1. You can change the default track priority or override the default for an individual track port:

- To change the default track priority, use the **backup track-priority** command, described below.
- To override the default track priority for a specific track port, use the **track-port** command. Refer to Specifying a track port on page 540.

To change the track priority, enter a command such as the following at the configuration level for the VRID.

```
device(config-vlan-200-vrid-1)# backup track-priority 2
```

Syntax: [no] backup [**priority value**] [**track-priority value**]

Specifying a track port

You can configure the VRID on one interface to track the link state of another interface on the device. This capability is useful for tracking the state of the exit interface for the path for which the VRID is providing redundancy. Refer to VSRP priority calculation on page 525.

To configure a VRID to track an interface, enter a command such as the following at the configuration level for the VRID.

```
device(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** command with the *num* variable.

Disabling or re-enabling Backup preemption

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the VRID. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the VRID.

Preemption is especially useful for preventing flapping in situations where there are multiple Backups and a Backup with a lower priority than another Backup has assumed ownership, because the Backup with the higher priority was unavailable when ownership changed.

If you enable the non-preempt mode (thus disabling the preemption feature) on all the Backups, the Backup that becomes the Master following the disappearance of the Master continues to be the Master. The new Master is not preempted.

To disable preemption on a Backup, enter a command such as the following at the configuration level for the VRID.

```
device(config-vlan-200-vrid-1)# non-preempt-mode
```

Syntax: [no] non-preempt-mode

Displaying VSRP information

You can display the following VSRP information:

- Configuration information and current parameter values for a VRID or VLAN
- The interfaces on a VSRP-aware device that are active for the VRID

Displaying VRID information

To display VSRP information, enter the following command.

```
device# show vsrp vrid 10
    VLAN 100
     Auth-type no authentication
     VRID 100
     ========
      State           Administrative-status Advertise-backup Preempt-mode
      Master          Enabled               Disabled         True
      Parameter        Configured Current    Unit/Formula
      Priority         100        100        (100-0)*(3.0/3.0)
      Hello-interval   1          1          sec/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
      Member ports:      ethe 1/1 ethe 2/1 ethe 2/10
      Operational ports: ethe 1/1 ethe 2/1 ethe 2/10
```

On a devices where the VSRP Fast Start feature is enabled.

```
device(config-vlan-100-vrid-100)#show vsrp vrid 100
VLAN 100
  auth-type no authentication
  VRID 100
  ========
   State      Administrative-status Advertise-backup Preempt-mode
   master     enabled               disabled         true
   Parameter        Configured Current    Unit/Formula
   priority         100        50         (100-0)*(2.0/4.0)
   hello-interval   1          1          sec/1
   dead-interval    3          3          sec/1
   hold-interval    3          3          sec/1
   initial-ttl      2          2          hops
   next hello sent in 00:00:00.3
   Member ports:      ethe 2/5 to 2/8
   Operational ports: ethe 2/5 ethe 2/8
   Forwarding ports:  ethe 2/5 ethe 2/8
   Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

Syntax: show vsrp [**vrid** *num* | **vlan** *vlan-id*]

This display shows the following information when you use the **vrid***num* or **vlan***vlan-id* parameter. For information about the display when you use the **aware** parameter, refer to Displaying the active interfaces for a VRID on page 544.

TABLE 66 CLI display of VSRP VRID or VLAN information

This field...	Displays...
Total number of VSRP routers defined	The total number of VRIDs configured on this device.
VLAN	The VLAN on which VSRP is configured.
auth-type	The authentication type in effect on the ports in the VSRP VLAN.
VRID parameters	
VRID	The VRID for which the following information is displayed.
state	This device's VSRP state for the VRID. The state can be one of the following: • initialize - VSRP is not enabled on the VRID. If the state remains "initialize" after you enable VSRP on the VRID, make sure that the VRID is also configured on the other routers Routing Switches and that the routers Routing Switches can communicate with each other. **NOTE** If the state is "initialize" and the mode is incomplete, make sure you have specified the IP address for the VRID. • standby - This device is a Backup for the VRID. • master - This device is the Master for the VRID.
Administrative-status	The administrative status of the VRID. The administrative status can be one of the following: • disabled - The VRID is configured on the interface but VSRP or VRRP-E has not been activated on the interface. • enabled - VSRP has been activated on the interface.
Advertise-backup	Whether the device is enabled to send VSRP Hello messages when it is a Backup. This field can have one of the following values: • disabled - The device does not send Hello messages when it is a Backup. • enabled - The device does send Hello messages when it is a Backup.

TABLE 66 CLI display of VSRP VRID or VLAN information (continued)

This field...	Displays...
Preempt-mode	Whether the device can be pre-empted by a device with a higher VSRP priority after this device becomes the Master. This field can have one of the following values: • disabled - The device cannot be pre-empted. • enabled - The device can be pre-empted.
NOTE For the following fields: • Configured - indicates the parameter value configured on this device. • Current - indicates the parameter value received from the Master. • Unit - indicates the formula used tor calculating the VSRP priority and the timer scales in effect for the VSRP timers. A timer's true value is the value listed in the Configured or Current field divided by the scale value.	
priority	The device's preferability for becoming the Master for the VRID. During negotiation, the Backup with the highest priority becomes the Master. If two or more Backups are tied with the highest priority, the Backup interface with the highest IP address becomes the Master for the VRID.
hello-interval	The number of seconds between Hello messages from the Master to the Backups for a given VRID.
dead-interval	The configured value for the dead interval. The dead interval is the number of seconds a Backup waits for a Hello message from the Master for the VRID before determining that the Master is no longer active. If the Master does not send a Hello message before the dead interval expires, the Backups negotiate (compare priorities) to select a new Master for the VRID. NOTE If the value is 0, then you have not configured this parameter.
hold-interval	The number of seconds a Backup that intends to become the Master will wait before actually beginning to forward Layer 2 traffic for the VRID. If the Backup receives a Hello message with a higher priority than its own before the hold-down interval expires, the Backup remains in the Backup state and does not become the new Master.
initial-ttl	The number of hops a Hello message can traverse after leaving the device before the Hello message is dropped. NOTE A Foundry MRP ring counts as one hop, regardless of the number of nodes in the ring.
next hello sent in	The amount of time until the Master's dead interval expires. If the Backup does not receive a Hello message from the Master by the time the interval expires, either the IP address listed for the Master will change to the IP address of the new Master, or this device itself will become the Master. NOTE This field applies only when this device is a Backup.
master router	The IP address of the master router.
Member ports	The ports in the VRID.
Operational ports	The member ports that are currently up.

TABLE 66 CLI display of VSRP VRID or VLAN information (continued)

This field...	Displays...
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.

```
device(config-vlan-200-vrid-1)# show vsrp aware
Aware port listing
VLAN ID  VRID  Last Port
100      1     3/2
200      2     4/1
```

Syntax: show vsrp aware

This display shows the following information when you use the **aware** parameter. For information about the display when you use the **vrid***num* or **vlan***vlan-id* parameter, refer to Displaying VRID information on page 541.

TABLE 67 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-Brocade or non-VSRP aware devices that are connected to a Brocade device that is the VSRP Master to quickly switch over to the new Master when a VSRP failover occurs

This feature causes the port on a VSRP Master to restart when a VSRP failover occurs. When the port shuts down at the start of the restart, ports on the non-VSRP aware devices that are connected to the VSRP Master flush the MAC address they have learned for the VSRP master. After a specified time, the port on the previous VSRP Master (which now becomes the Backup) returns back online. Ports on the non-VSRP aware devices switch over to the new Master and learn its MAC address.

Special considerations when configuring VSRP fast start

Consider the following when configuring VSRP fats start:

- VSRP is sensitive to port status. When a port goes down, the VSRP instance lowers its priority based on the port up fraction. (refer to VSRP priority calculation on page 525 for more information on how priority is changed by port status). Since the VSRP fast start feature toggles port status by bringing ports down and up it can affect VSRP instances because their priorities get reduced when a port goes down. To avoid this, the VSRP fast start implementation keeps track of ports that it brings down and suppresses port down events for these ports (as concerns VSRP).

- Once a VSRP restart port is brought up by a VSRP instance, other VSRP instances (in Master state) that have this port as a member do not go to forwarding immediately. This is a safety measure that is required to prevent transitory loops. This could happen if a peer VSRP node gets completely cut off from this node and assumed Master state. In this case, where there are 2 VSRP instances that are in Master state and forwarding, the port comes up and starts forwarding immediately. This would cause a forwarding loop. To avoid this, the VSRP instance delays forwarding.

Recommendations for configuring VSRP fast start

The following recommendations apply to configurations where multiple VSRP instances are running between peer devices sharing the same set of ports:

- Multiple VSRP instances configured on the same ports can cause VSRP instances to be completely cut off from peer VSRP instances. This can cause VSRP instances to toggle back and forth between master and backup mode. For this reason, we recommend that you configure VSRP fast start on a per port basis rather than for the entire VLAN.
- We recommend that VSRP peers have a directly connected port without VSRP fast start enabled on it. This allows protocol control packets to be received and sent even if other ports between the master and standby are down.
- The VSRP restart time should be configured based on the type of connecting device since some devices can take a long time to bring a port up or down (as long as several seconds). In order to ensure that the port restart is registered by neighboring device, the restart time may need to be changed to a value higher than the default value of 1 second.

Configuring VSRP fast start

The VSRP fast start feature can be enabled on a VSRP-configured device, either on the VLAN to which the VRID of the VSRP-configured device belongs (globally) or on a port that belongs to the VRID.

To globally configure a VSRP-configured device to shut down its ports when a failover occurs, then restart after five seconds, enter the following command.

```
device(configure)# vlan 100
device(configure-vlan-100)# vsrp vrid 1
device(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.

```
device(configure)# interface ethernet 1/1
device(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.

```
device(config-vlan-100-vrid-100)#show vsrp vrid 100
VLAN 100
  auth-type no authentication
  VRID 100
  ========
```

```
State      Administrative-status Advertise-backup Preempt-mode save-current
master     enabled               disabled         true         false
Parameter        Configured Current    Unit/Formula
priority         100        50         (100-0)*(2.0/4.0)
hello-interval   1          1          sec/1
dead-interval    3          3          sec/1
hold-interval    3          3          sec/1
initial-ttl      2          2          hops
next hello sent in 00:00:00.3
Member ports:      ethe 2/5 to 2/8
Operational ports: ethe 2/5 ethe 2/8
Forwarding ports:  ethe 2/5 ethe 2/8
Restart ports:     2/5(1) 2/6(1) 2/7(1) 2/8(1)
```

The "Restart ports:" line lists the ports that have the VSRP fast start enabled, and the downtime for each port. Refer to Displaying VRID information on page 541 to interpret the remaining information on the display.

VSRP slow start

In a VSRP configuration, if a Master router goes down, the Backup router with the highest priority takes over. When the Master comes back up again, it takes over from the Backup. By default, this transition from Backup back to Master takes place immediately. You can configure the VSRP slow start timer feature, which causes a specified amount of time to elapse between the time the Master is restored and when it takes over from the Backup. (This range is currently set to between 1 to 600 ticks (1/10 second to 60 seconds). This interval allows time for VSRP convergence when the Master is restored.

To set the VSRP slow start timer to 30 seconds, enter the following command.

```
device(config)# router vsrp
device(config-vsrp-router)# slow-start 300
```

Syntax: [no] slow-start ticks

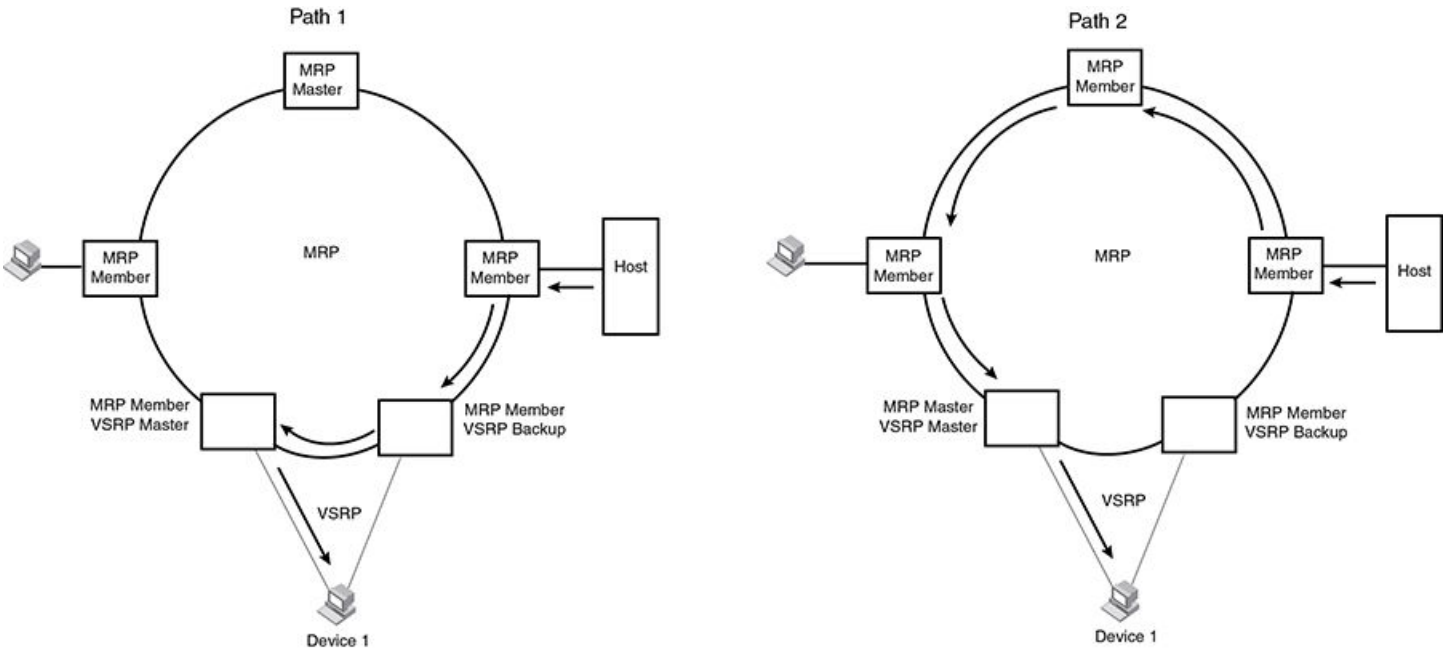
The *ticks* parameter can range is from 1 to 600 ticks (1/10 second to 60 seconds).

When the VSRP slow start timer is enabled, if the Master goes down, the Backup takes over immediately. If the Master subsequently comes back up again, the amount of time specified by the VSRP slow start timer elapses (in this example, 30 seconds) before the Master takes over from the Backup.

VSRP and Foundry MRP signaling

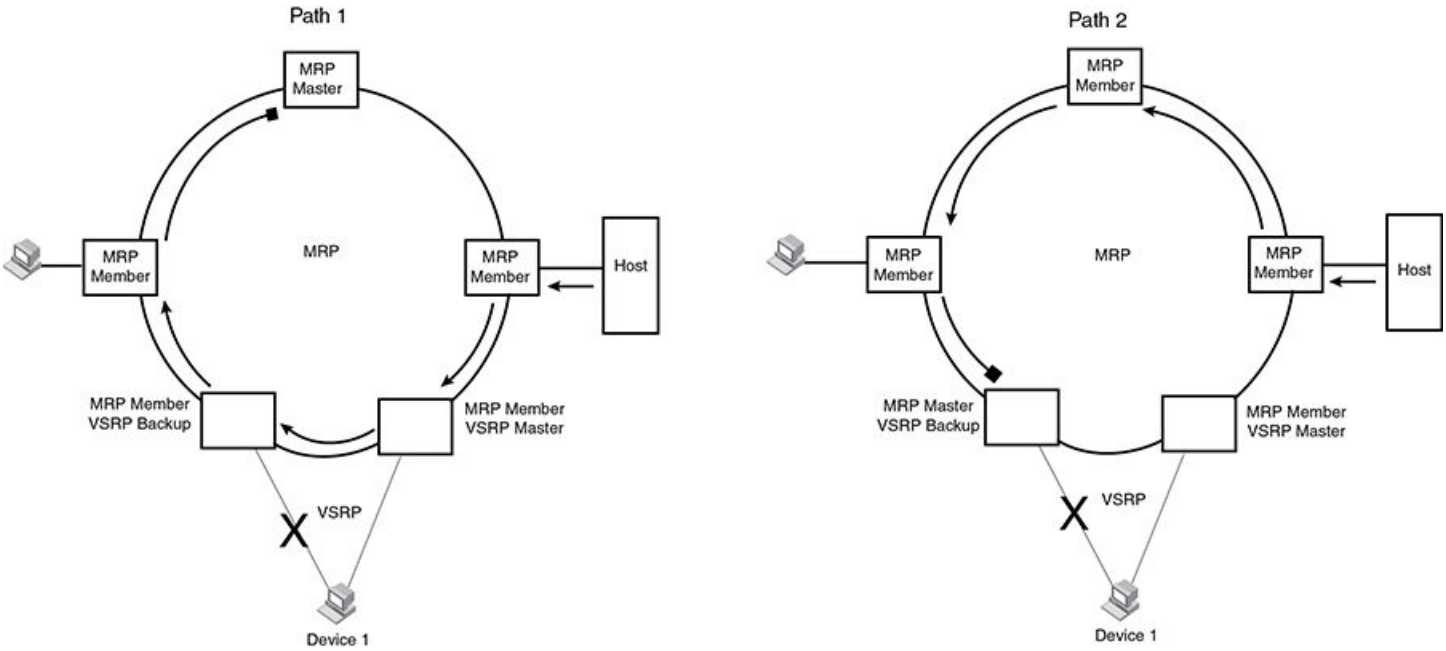
A device may connect to a Foundry MRP ring through VSRP to provide a redundant path between the device and the Foundry MRP ring. VSRP and Foundry MRP signaling, ensures rapid failover by flushing MAC addresses appropriately. The host on the Foundry MRP ring learns the MAC addresses of all devices on the Foundry MRP ring and VSRP link. From these MAC addresses, the host creates a MAC database (table), which is used to establish a data path from the host to a VSRP-linked device. Figure 159 below shows two possible data paths from the host to Device 1.

FIGURE 156 Two data paths from host on a Foundry MRP ring to a VSRP-linked device



If a VSRP failover from master to backup occurs, VSRP needs to inform Foundry MRP of the topology change; otherwise, data from the host continues along the obsolete learned path and never reach the VSRP-linked device, as shown in Figure 160.

FIGURE 157 VSRP on Foundry MRP rings that failed over

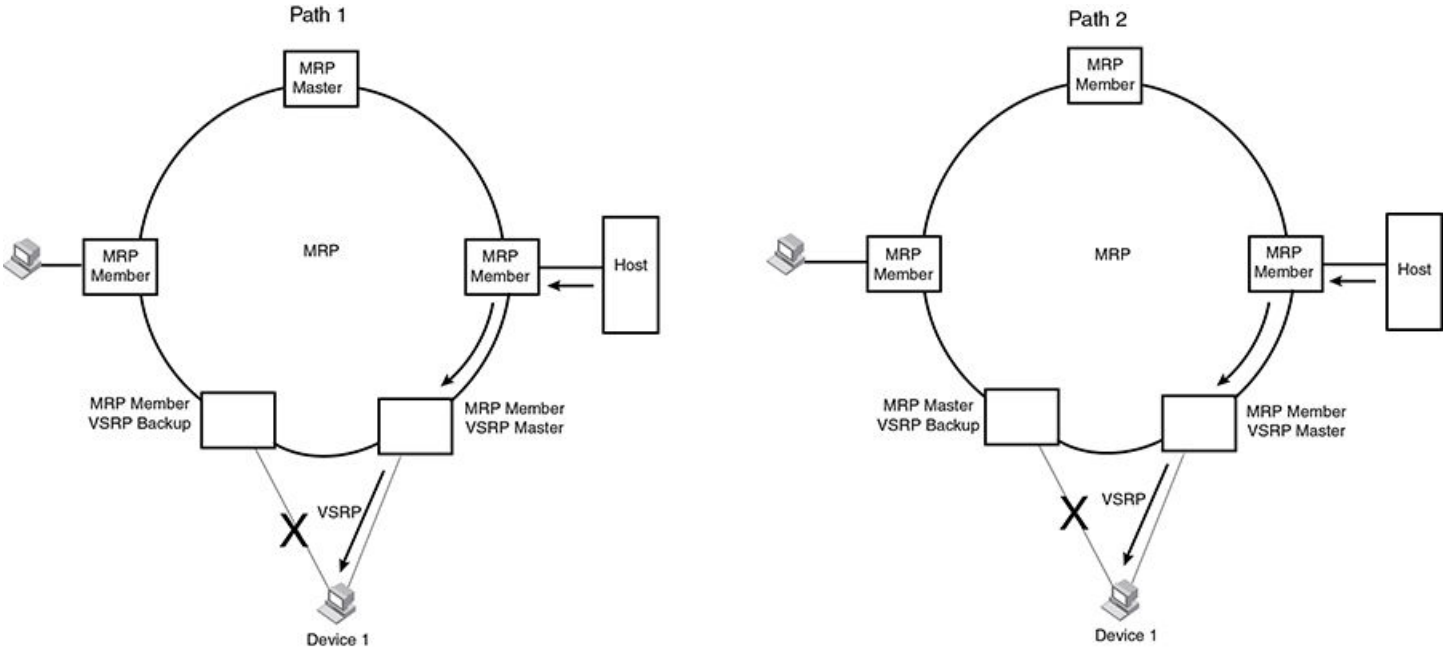


To ensure that Foundry MRP is informed of the topology change and to achieve convergence rapidly, a signaling process for the interaction between VSRP and Foundry MRP. When a VSRP node fails, a new VSRP master is selected. The new VSRP master finds all Foundry MRP instances impacted by the failover. Then each Foundry MRP instance does the following:

- The Foundry MRP node sends out a Foundry MRP PDU with the mac-flush flag set three times on the Foundry MRP ring.
- The Foundry MRP node that receives this Foundry MRP PDU empties all the MAC address entries from its interfaces that participate on the Foundry MRP ring.
- The Foundry MRP node then forwards the Foundry MRP PDU with the mac-flush flag set to the next Foundry MRP node that is in forwarding state.

The process continues until the Master Foundry MRP node's secondary (blocking) interface blocks the packet. Once the MAC address entries have been flushed, the MAC table can be rebuilt for the new path from the host to the VSRP-linked device (Figure 161).

FIGURE 158 New path established



There are no CLI commands used to configure this process.

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 Brocade device is deployed in a Metro network and provides forwarding for two Foundry MRP rings that each contain 128 VLANs, you can configure a topology group for each ring. If a link failure in a ring causes a topology change, the change is applied to all the VLANs in the ring's topology group. Without topology groups, you would need to configure a separate ring for each VLAN.

You can use topology groups with the following Layer 2 protocols:

- STP
- Foundry MRP
- VSRP
- RSTP
- Ethernet RIng Protection (ERP)

Master VLAN and member VLANs

Each topology group contains a master VLAN and can contain one or more member VLANs and VLAN groups. A definition for each of these VLAN types follows:

- **Master VLAN** - The master VLAN contains the configuration information for the Layer 2 protocol. For example, if you plan to use the topology group for Foundry MRP, the topology group's master VLAN contains the ring configuration information.
- **Member VLANs** - The member VLANs are additional VLANs that share ports with the master VLAN. The Layer 2 protocol settings for the ports in the master VLAN apply to the same ports in the member VLANs. A change to the master VLAN's Layer 2 protocol configuration or Layer 2 topology affects all the member VLANs. Member VLANs do not independently run a Layer 2 protocol. VPLS VLANs can become member VLANs within a topology group.
- **Member VLAN groups** - A VLAN group is a named set of VLANs. The VLANs within a VLAN group have the same ports and use the same values for other VLAN parameters.

When a Layer 2 topology change occurs on a port in the master VLAN, the same change is applied to that port in all the member VLANs that contain the port. For example, if you configure a topology group whose master VLAN contains ports 1/1 and 1/2, a Layer 2 state change on port 1/1 applies to port 1/1 in all the member VLANs that contain that port. However, the state change does not affect port 1/1 in VLANs that are not members of the topology group.

Master VLANs and customer VLANs in Foundry MRP

A topology group enables you to control forwarding in multiple VLANs using a single instance of a Layer 2 protocol such as Foundry MRP. For more information on topology group and Foundry MRP, refer to the *VLANs* chapter.

Control ports and free ports

A port in a topology group can be a control port or a free port:

- A **Control port** is a port in the master VLAN and, therefore, is controlled by the Layer 2 protocol configured in the master VLAN. The same port in all the member VLANs is controlled by the master VLAN's Layer 2 protocol. Each member VLAN must contain all of the control ports. All other ports in the member VLAN are "free ports."
- **Free ports** are not controlled by the master VLAN's Layer 2 protocol. The master VLAN can contain free ports. (In this case, the Layer 2 protocol is disabled on those ports.) In addition, any ports in the member VLANs that are not also in the master VLAN are free ports.

  NOTE
  Because free ports are not controlled by the master port's Layer 2 protocol, they are assumed always to be in the forwarding state, when enabled.

Configuration considerations

The configuration considerations are as follows:

- You can configure up to 255 topology groups. Each group can control up to 4000 VLANs. A VLAN cannot be controlled by more than one topology group.
- The topology group must contain a master VLAN and can also contain individual member VLANs, VLAN groups, or a combination of individual member VLANs and VLAN groups. Therefore, you configure the master VLAN and member VLANs or member VLAN groups before you configure a topology group.
- After you add a VLAN as a member of a topology group, the device deletes all the Layer 2 protocol information on that VLAN.
- If you add a new master VLAN to a topology group that already has a master VLAN, the new master VLAN replaces the older master VLAN. All member VLANs and VLAN groups follow the Layer 2 protocol settings of the new master VLAN.
- If you remove the master VLAN (by entering the **no master-vlan** command with the *vlan-id* variable), the software selects the new master VLAN from member VLANs. A new candidate master VLAN will be in configured order to a member VLAN so that the first added member VLAN will be a new candidate master VLAN. Once you save and reload, a member VLAN with the youngest VLAN ID will be a new candidate master. The new master VLAN inherits the Layer 2 protocol settings of the older master VLAN.
- After you add a VLAN or VLAN group as a member of a topology group, all the Layer 2 protocol configuration information for the VLAN or group is deleted. For example, if STP is configured on a VLAN and you add the VLAN to a topology group, the STP configuration is removed from the VLAN. After you add the VLAN to a topology group, the VLAN uses the Layer 2 protocol settings of the master VLAN. If you remove a member VLAN or VLAN group from a topology group, you need to reconfigure the Layer 2 protocol information in the VLAN or VLAN group.
- On platforms where the Ethernet Service Instance (ESI) framework is supported, master VLANs in a topology group must either be in the default ESI or within the same ESI. Master and member VLANs cannot span multiple ESIs.

Configuring a topology group

To configure a topology group, enter commands such as the following.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan 2
device(config-topo-group-2)# member-vlan 3
device(config-topo-group-2)# member-vlan 4
device(config-topo-group-2)# member-vlan 5
device(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 **topology-group** command creates a topology group. The *group-id* parameter assigns an ID 1 to 255 to the topology group.

Syntax: [no] master-vlan vlan-id

This command adds the master VLAN to the topology group. The VLAN must already be configured. Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

> NOTE
> When a port is added to a master VLAN, it will be added as a free port. Similarly when a port has to be removed from master VLAN, first disable any the Layer 2 protocol on the port, then remove the port from the master VLAN.

Syntax: [no] member-vlan vlan-id

This command adds a member VLAN to the topology group. The VLAN must already be configured.

Syntax: [no] member-group num

This command adds a VLAN group to the topology group. The *num* specifies a VLAN group ID. The VLAN group must already be configured.

Adding VPLS VLANs to topology groups

To add *single-tagged or untagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following example.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan 2
device(config-topo-group-2)# member-vlan vpls id 34 vlan 42 to 45
```

To add *dual-tagged* VPLS VLANs as member VLANs to a topology group, use the **member-vlan vpls** command as shown in the following configuration example.

```
device(config)# topology-group 1
device(config-topo-group-1)# master-vlan 10
device(config-topo-group-1)# member-vlan 20
device(config-topo-group-1)# member-vlan vpls id 5 vlan 300 inner-vlan 20 to 25
```

Syntax: [no] member-vlan vpls [**id** *vpls-id* | **name** *vpls-name* } **vlan** *vlan-id* [**to** *vlan-id*]

OR

Syntax: [no] member-vlan vpls [**id** *vpls-id* | **name** *vpls-name*] **vlan** *vlan-id* [**inner-vlan** *inner-vlan-id* [**to** *inner-vlan-id*]]

The **id** option allows you to specify the VPLS instance that you are configuring into the topology group by using the VPLS ID of the instance. A value in the range of 1 - 4294967294 can be entered for VPLS ID.

The **name** option allows you to specify the VPLS instance that you are configuring into the topology group by using the name of the instance.

The *vlan-id* variable is used with the **vlan** keyword to specify the VPLS VLAN being configured into topology group. You can specify multiple *vlan-id* values or specify a range of VLANs using the **to** option.

The **inner-vlan** option allows you to specify a VPLS dual-tagged (double-tagged) VLAN configuration.

NOTE
The **inner-vlan** option does not allow both outer VLAN ranges and inner VLAN ranges for a given VPLS instance. Once an outer VLAN range is specified, the inner VLAN option is not allowed. However, if a single outer VLAN is specified, the inner VLAN option and range is allowed.

NOTE
You cannot delete a topology master VLAN if the topology group has only VPLS VLAN members and no Layer 2 VLAN members because the normal procedure for deleting a topology master VLAN is to elect another Layer 2 VLAN as the new master. Because a VPLS VLAN cannot be a master VLAN, you must have at least one Layer 2 VLAN as a member. If it does not currently exist, you must add a Layer 2 VLAN before deleting a topology master.

NOTE
A maximum of 4000 VPLS member VLANs can be added to a topology group.

Topology group support within an ESI

Topology groups can be configured with VLANs that are part of a user-defined ESI. (Consult Topology Groups on page 177 to see which platform supports topology groups within an ESI.) When you configure topology groups in such a scenario, both the master and member VLANs must be part of the same ESI. If an ESI is not specified, the system assumes a reference to the default ESI. Below is an example of configuring topology groups with VLANs that are part of a user-defined ESI.

```
device(config)# topology-group 2
device(config-topo-group-2)# master-vlan service-esi 2
device(config-topo-group-2)# member-vlan service-esi 3
device(config-topo-group-2)# member-vlan service-esi 4
device(config-topo-group-2)# member-vlan service-esi 5
device(config-topo-group-2)# member-group service-esi 2
```

The commands configure topology group 2 and add the following to it:

- VLAN 2 in ESI "service-esi" as master VLAN
- VLANs 3, 4, and 5 in ESI "service-ESI" as member VLANs
- Member VLAN group 2

Syntax: [no] topology-group group-id

This command creates a topology group. The *group-id* parameter assigns an ID in the range 1 to 255 to the topology group.

Syntax: [no] master-vlan esi-name vlan-id

This command adds the master VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name". Make sure all the Layer 2 protocol settings in the VLAN are correct for your configuration before you add the VLAN to the topology group. A topology group can have only one master VLAN.

Syntax: [no] member-vlan esi-name vlan-id

This command adds a member VLAN in ESI identified by the VLAN ID to the topology group. The VLAN must already be configured in the ESI "esi-name".

Syntax: [no] member-group esi-name num

This command adds a VLAN group in ESI identified by "esi-name" to the topology group. The *num* specifies a VLAN group ID. The VLAN group must already be configured.

Displaying topology group information

This section contains examples of the **show topology-group** command output. Support for topology groups within an ESI is supported on a minority of platforms (listed in Displaying topology group information on a Brocade NetIron CES Series device on page 183), so its example appears at the end of this section.

Displaying topology group information on a Brocade NetIron XMR Series or Brocade NetIron MLX Series device

The **show topology-group** command offers a choice between one of two mandatory parameters. The command syntax (on a Brocade NetIron XMR Series or Brocade NetIron MLX Series device) is as follows.

Syntax: show topology-group *group-id* | **hw-index-table** [*hw-index*]

The first example in this section utilizes the first possible mandatory parameter, *group-id*. The second example utilizes the second possible mandatory parameter, **hw-index-table**, along with an optional variable, a hardware index number.

Display topology group information by using a Group ID

To display topology group information for group 10, enter the **show topology-group** command.

```
device#show topology-group 10
Topology Group 10
=================
Topo HW Index   : 0
Master VLAN     : 10
VPLS VLAN exist : TRUE
Member VLAN     : 20
Member Group  : None
Control Ports : ethe 3/11 to 3/12 ethe 3/15 to 3/16
Free Ports :
```

Syntax: show topology-group group-id

This display shows the following information:

TABLE 68 CLI display of topology group information

This field...	Displays...
Topology Group	The ID of the topology group. The range for *group-id* is 1 - 256.
Topo HW Index	A topology hardware index is a unique hardware ID that is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The range for *hw-index* is 0 - 511. (The **show topology-group hw-index-table** command output shows the mapping of a topology hardware index to a VLAN.)
Master-VLAN	The master VLAN for the topology group. The settings for STP, Foundry MRP, ERP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
VPLS VLAN exist	Indicates whether the topology group has one or more VPLS VLANs as a topology group member. The content of this field is TRUE or FALSE.
Member-VLAN	The VLAN ID of the member of the topology group.
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.

TABLE 68 CLI display of topology group information (continued)

This field...	Displays...
Free ports	A list of all free ports in the topology group. A free port is not controlled by the Layer 2 protocol information in the master VLAN. In the example screen output, the absence of any number indicates that no ports are free.

Display topology group information by using hardware index table numbers

Display the information for hardware index table 0.

```
device#show topology-group hw-index-table  0
Total Instances : 512
Free Instances  : 511
Topo HW Index    Vlan ID
-------------    -------
0                10
```

Syntax: show topology-group hw-index-table [**hw-index**]

The range for *hw-index* is 0 - 511. If you do not specify a number for *hw-index*, the output screen lists all entries.

TABLE 69 Topology group information with hardware index table

This field...	Displays...
Total Instances	Total number of topology hardware indexes that have been initialized in the system.
Free Instances	Number of free topology hardware indexes that are left in the system.
Topology HW Index	A topology hardware index is assigned to a VLAN when a Layer 2 protocol is configured on the VLAN. The VLAN that runs the Layer 2 protocol could be a standalone Layer 2 VLAN or a master VLAN under a topology group. The **show topology-group hw-index-table** command output shows the mapping of a topology hardware index to a VLAN. The range for is 0 - 511. In the example, hardware index table 0 is mapped to the VLAN with an ID of 10.
VLAN ID	The ID of the port-based VLAN that owns the protocol instance on that VLAN. VLAN 1 is the default VLAN. If you have not configured port-based VLANs on a device, all protocol information is for VLAN 1.

Displaying topology group information on a Brocade NetIron CES Series device

To display topology group information within an ESI, enter the **show topology-group** command, as in the following example.

```
device(config)# show topology-group 3
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 70 CLI display of topology group information

This field...	Displays...
master-vlan	The master VLAN for the topology group. The settings for STP, Foundry MRP, RSTP, or VSRP on the control ports in the master VLAN apply to all control ports in the member VLANs within the topology group.
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: • Foundry MRP • STP • RSTP • VSRP • ERP
Per vlan free ports	The ports that are not controlled by the Layer 2 protocol information in the master VLAN.

Multi-Chassis Trunking (MCT)

About Multi-Chassis Trunk (MCT)

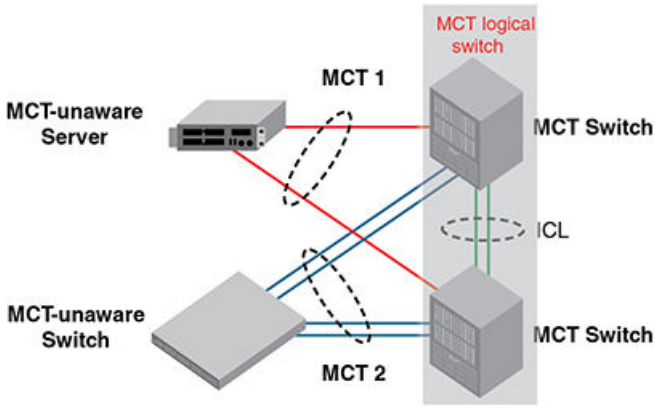
A Multi-Chassis Trunk (MCT) is a trunk that initiates at a single MCT-unaware server or switch and terminates at two MCT-aware switches.

Link Aggregation (LAG) trunks provide link level redundancy and increased capacity. However, LAG trunks do not provide switch-level redundancy. If the switch to which the LAG trunk is attached fails, the entire LAG trunk loses network connectivity. With MCT, member links of the LAG are connected to two chassis. The MCT switches may be directly connected using an Inter-Chassis Link (ICL) to enable data flow and control messages between them. In this model, if one MCT switch fails, a data path will remain through the other switch.

In an MCT scenario, all links are active and can be load shared to increase bandwidth. In addition, traffic restoration can be achieved in milliseconds after an MCT link failure or MCT switch failure.

MCT is designed to increase network resilience and performance.

FIGURE 159 Chassis trunk example



MCT Benefits

MCT provides the following benefits:

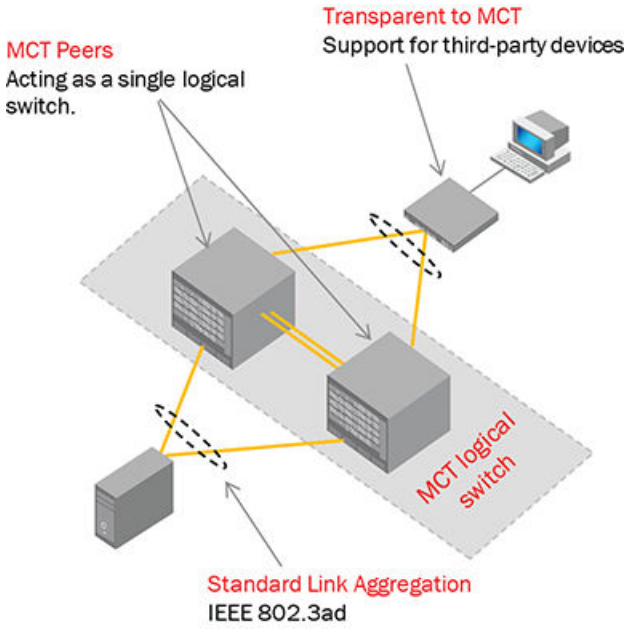
- Provides link-level and switch-level redundancy.
- Provides increased capacity because it utilizes all links (including redundant ones) for traffic transport. This contrasts with the use of the Spanning Tree Protocol, which does not use redundant links for transporting traffic.
- Provides traffic restoration in tens of milliseconds in case of link or switch failures.
- Allows servers and switches to have redundant connections to two switches and to fully utilize all links (including redundant ones) for traffic transport.
- Allows servers and switches to use standard link aggregation (802.3ad) to connect to redundant switches.
- MCT is easily deployed while enhancing existing multilayer switching without fundamentally changing the architecture.

How MCT works

The MCT is made up of the following:

- Sub-second failover in the event of a link, module, switch fabric, control plane, or node failure
- Layer 2 and Layer 3 forwarding (when using fast path forwarding) at the first hop regardless of VRRP-E state.
- Flow based load balancing rather than VLANs sharing across network links
- Ability to provide the resiliency regardless of the traffic type layer 3, layer 2 or non-IP (legacy protocols).
- Interaction with MRP to build larger resilient Layer 2 domains
- Enhancement to Link Aggregation Groups
- Provides nodal redundancy in addition to link and modular redundancy
- Operates at the physical level to provide sub-second failover

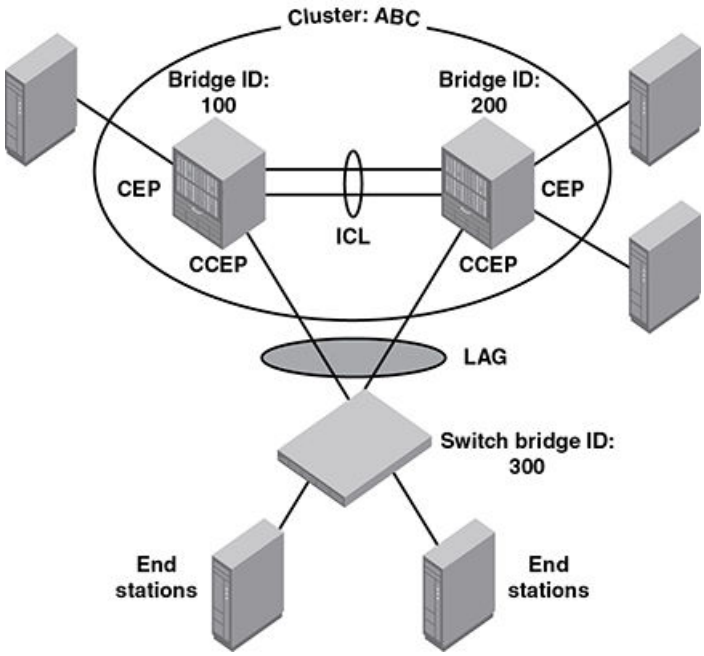
FIGURE 160 How MCT works



MCT components

To properly understand MCT, consider Figure 164, which shows an example of MCT deployment, functions and features.

FIGURE 161 MCT Components



MCT terminology

- MCT peer switches: A pair of switches connected as peers through the ICL. The LAG interface is spread across two MCT peer switches and acts as the single logical endpoint to the MCT client.
- MCT client: The MCT client is the device that connects with MCT peer switches through an IEEE 802.3ad link. It can be a switch or an endpoint server host in the single-level MCT topology or another pair of MCT switches in a multi-tier MCT topology.
- MCT Inter-Chassis Link (ICL): A single-port or 1 GbE or 10 GbE interface between the two MCT peer switches. This link is typically a standard IEEE 802.3ad Link Aggregation interface. ICL ports should not be untagged members of any VLAN. The ICL is a tagged Layer 2 link, which carries packets for multiple VLANs. MCT VLANs are the VLANs on which MCT clients are operating. On the Brocade NetIron XMR or Brocade NetIron MLX series, non-MCT VLANs can coexist with MCT VLANs on the ICL. However, on the Brocade NetIron CES 2000 Series and Brocade NetIron CER 2000 Series, only MCT VLANs are carried over the ICL. The Brocade NetIron devices support up to 4096 ICL VLANs for a cluster.
- MCT Cluster Communication Protocol (CCP): A Brocade proprietary protocol that provides reliable, point-to-point transport to synchronize information between peers. CCP comprises two main components: CCP peer management and CCP client management. CCP peer management deals with establishing and maintaining a TCP transport session between peers, while CCP client management provides event-based, reliable packet transport to CCP peers.
- MCT Cluster Client Edge Port (CCEP): A physical port on one of the MCT peer switches that is a member of the LAG interface to the MCT client. To have a running MCT instance, at least one Link Aggregation Interface is needed with a member port on each peer switch.
- MCT Cluster Edge Port (CEP): A port on MCT peer switches that is neither a Cluster Client Edge Port nor an ICL port.
- MCT VLANs: VLANs on which MCT clients are operating. These VLANs are explicitly configured in the MCT configuration by the user.

  NOTE
  For MCT VLANs, MAC learning is disabled on the ICL ports, while MAC learning is enabled on ICL port for non-MCT VLANs.

- MCT session VLANs: The VLAN used by the cluster for control operations. CCP runs over this VLAN. The interface can be a single link or a LAG port. If it is a LAG port, it should be the primary port of the LAG.

  NOTE
  The MCT session VLAN's subnet will not be distributed in routing protocols using redistribute commands.

- RBridge ID: RBridge ID is a value assigned to MCT nodes and clients to uniquely identify them, and helps in associating the source MAC address with an MCT node.
- MAC Database Update Protocol (MDUP)
- CL: Cluster Local MACs
- CCL: Cluster Client Local MACs
- CR: Cluster Remote MACs
- CCR: Cluster Client Remote MACs
- CCRR: Cluster Client RBridge Reachability
- MDB: MAC Database. The MDB can have multiple MAC entries for the same address.
- FDB: Forwarding MAC Database. The FDB will have the best MAC address only installed.

NOTE
BFD sessions configured with lower timer values may exhibit flaps when configured alongside MACSec on same line card. This issue is a known limitation. However, the MCT sessions are stable with 1 second for 3 tries in such scenarios.

Dynamic LAGs

MCT Client creates a single dynamic LAG towards the MCT nodes. For MCT nodes the dynamic Lag consists of two LAGs, each is configured on one of the MCT devices. A dynamic LAG runs Link Aggregation Control Protocol (LACP).

For the two dynamic LAGs of the MCT to behave as a single LAG from the MCT client's perspective, both of the dynamic LAGs should have the same LACP system ID and key, referred to as the MCT system ID and MCT key. In a system configuration with multiple MCT peers, the LACP system priority on both the MCT nodes should be same.

The MCT system ID and MCT key is uniquely defined for one MCT. They have the following attributes:

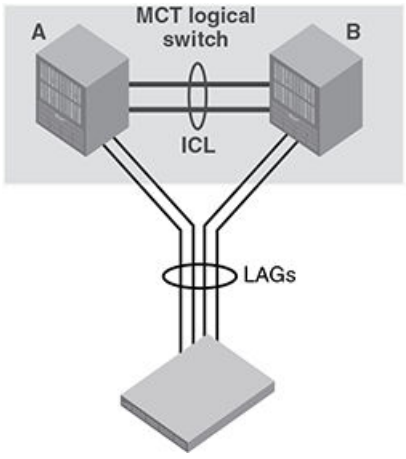
- MCT base system id = 0180.c200.0000
- MCT system id = MCT base system ID + cluster ID
- The cluster ID is user configurable on each MCT peer and unique across the MCT system
- MCT base key = 30000
- MCT LAG Group ID = MCT base key + client bridge ID

NOTE
Each MCT node has a unique cluster ID and an MCT client ID. If you need to configure the trunk threshold value for a LAG, make sure that the threshold value is less than the number of ports in the LAG.

MCT peers

Each MCT physical node, A and B, will act as an MCT peer and they are connected using an ICL. The pair of MCT peers will act as one logical switch for the access switch or server so that the MCT pair can connect using standard LAG to them. This is illustrated in MCT peers.

FIGURE 162 MCT peers



ICL traffic handling

An ICL link on the Brocade device can be a single port or a static or LACP LAG. Non-MCT VLANs can co-exist with MCT VLANs on the ICL only on the Brocade NetIron MLX Series and NetIron XMR . For MCT VLANs, MAC learning is disabled on ICL ports.

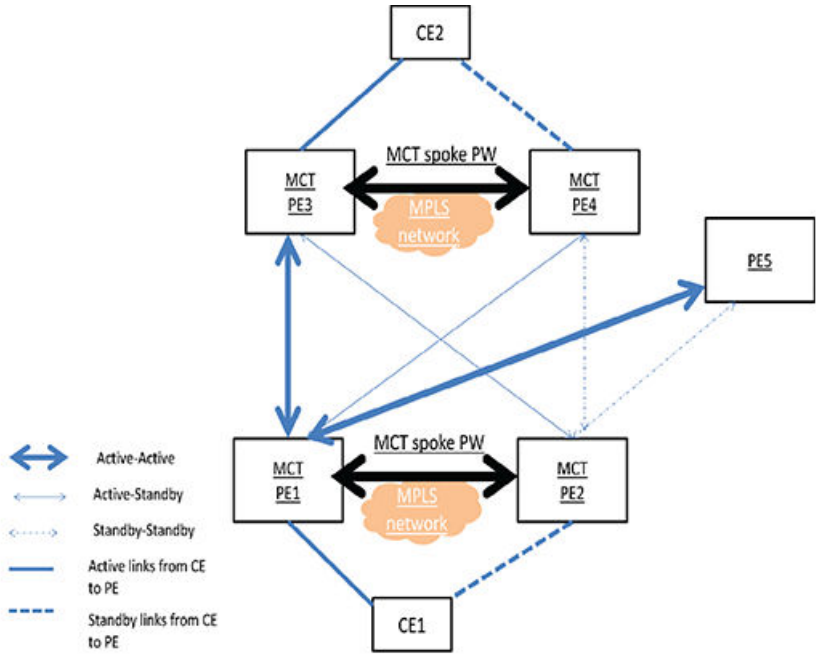
MCT Active-Passive mode

Using the MCT Active-Active mode, both MCT nodes will be Active forwarding the traffic coming from the client nodes on the CCEP links.

Using the MCT Active-Passive mode, the CCEP ports on one of the MCT nodes will be made Passive by blocking, and the other MCT node will be Active.

To enable MCT Active-Passive mode to work, both MCT nodes must be configured with the ACTIVE-PASSIVE feature. If only one node is configured, then cluster CCP session will not come up.

FIGURE 163 MCT Active-Passive topology



MCT Active-Passive mode is an example of an MCT Active-Passive topology. PE1 and PE2 are nodes of cluster, MCT1. PE3 and PE4 are nodes of another MCT cluster, MCT2.

- CE1 and CE2 are clients of MCT1 and MCT2 respectively.
- Data traffic will be forwarded on the Active-Active links between the PEs.
- Data traffic will be forwarded on the Active links from the CE to PE.

Multicast snooping over MCT

To support multicast snooping over MCT for Brocade devices, the ICL port is used to synchronize the following information between the cluster switches using MDUP:

- MAC - forward entries (mcache entries on MCT VLAN).
- IGMP or MLD Join or Leave (control packets on MCT VLAN).
- PIM-SM Join or Prune (control packets on MCT VLAN).

IGMP or MLD snooping

Each cluster switch in the MCT VLAN can be configured either as active or passive. There is no restriction for cluster switches to run Active-Active or Passive-Passive configuration.

Implementation of MCT is exactly the same for both IGMP or MLD Snooping, so all further details are explained only with reference to IGMP.

IGMP and MLD Control Packet Processing on MCT cluster switches

For IGMP reports and leaves,

- Native Packets coming to CPU from CCEP endpoints will be encapsulated in MDUP header and sent across ICL link through the TCP connection to the remote cluster peer, along with required control information in the header indicating the packet was received from a CCEP link, VLAN and client RBridgeID.
- Native Packets coming to CPU from CEP endpoint will be encapsulated in MDUP header and sent across ICL link through the TCP connection to the remote cluster peer, along with required control information in the header indicating the packet was received from a CEP link, VLAN and peer RBridgeID.
- Native Packets coming to CPU from ICL will not be processed to learn the group member. Only the MDUP messages that are sent through the TCP connection established between ICL links will be used to build the group OIF list.
- Packets coming from CEP and CCEP will be forwarded to router ports that are CEP, CCEP and ICL.
- Packets coming from ICL will be forwarded to router ports that are CEP. Packets will not be forwarded to router ports that are CCEP unless the peer CCEP ports are down.

For IGMP queries,

- Queries coming from CEP and CCEP will be forwarded to non-querier port that is CEP, CCEP and ICL.
- Queries coming from ICL will be forwarded to non-querier port that is CEP. Packet will not be forwarded to non-querier port that is CCEP unless the peer CCEP port is down.

Forwarding entries for multicast snooping

TABLE 71 Forwarding entries (,G)

Event	MCT-1	MCT-2
No-Join	(*,G)->blackhole	(*,G)->blackhole
(S,G)Join on (MCT-1)CEP	(*,G)->CEP - sources maintained on a port hash-list.	(*,G)->ICL
(S,G)Join on (MCT-2)CEP	(*,G)->ICL	(*,G)->CEP
(S,G)Join on (MCT-1)CCEP	(*,G)->CCEP , ICL]	(*,G)->CCEP, ICL
(S,G)Join on (MCT-2)CCEP	(*,G)->CCEP, ICL	(*,G)->CCEP, ICL

TABLE 72 Forwarding entries (S,G)

Event	MCT-1	MCT-2
No-Join	(S,G)->blackhole	(S,G)->blackhole
Join (MCT-1)CEP	(S,G)->CEP	(S,G)->ICL
Join on (MCT-2)CEP	(S,G)->ICL (S,G)->CEP	(S,G)->CEP
Join (MCT-1)CCEP	(S,G)->CCEP, ICL	(S,G)->CCEP, ICL
Join (MCT-2)CCEP	(S,G)->CCEP, ICL	(S,G)->CCEP, ICL

L2 protocol packet handling

The default behavior is to forward or flood STP and RSTP BPDU packets.

If the **no cluster-l2protocol-forward** command is configured on global basis or **cluster-l2protocol-forward** disable is configured on a port, the STP protocol packets coming on the ICL ports of MCT VLANs are dropped.

All other L2 protocol packets will be flooded on the MCT VLANs or dropped. The **cluster-l2protocol-forward** command is not applicable to these protocol packets. It only applies to STP or RSTP BPDU packets on the ICL ports only.

Forwarding broadcast, multicast and unknown unicast traffic

Traffic received from non-ICL ports is forwarded the same way as non-MCT devices.Traffic received from an ICL port is not forwarded to the CCEP port if the peer MCT switch has the reachability to the same cluster client.

Both the MCT nodes must be multicast enabled for multicast routing on MCT.

NOTE
When there is a double failure, the forwarding behavior will be unpredictable and there may be a complete traffic loss. For example, when both the ICL cluster link and any one leg of the client CCEP link fail. From the physical topology perspective, it may appear like a path is available, while traffic may not be forwarded.

NetIron CES and NetIron CER forwarding

The ICL port must belong to VLANs that are cluster member VLANs.

Syncing interface MACs to peer MCT devices

The MCT device uses an interface MAC to identify the packets that are addressed to the switch. Such packets may be received by a peer MCT device. The peer MCT device switches packets over the ICL to the local MCT switch to be routed properly.

MCT L2 protocols

When configuring L2 protocols, you should consider the following.

MRP

- An ICL interface can not be configured as MRP secondary interface and vice-versa as the ICL cannot be BLOCKING.
- MRP can not be enabled on MCT CCEP port and vice-versa.

G.8032

- If the port is an ERP interface, it can not be enabled as a CCEP port.
- If the interface is ICL interface it can not be configured with MS, FS or RPL.
- G.8032 and MCT are not supported together.

STP

- The STP algorithm has been modified such that ICL never goes to blocking. ICL guard mechanism ensures that if ICL is going in blocking state then the port on which the superior BPDUs are being received is moved to BLOCKING state and ICL guard timer starts running on it. This timer runs as long as Superior BPDUs are received on this interface. As long as this timer runs on an interface the Superior BPDUs are dropped.
- The modified STP algorithm also ensures that the CCEP interface state on both the MCT peers is same.
- The CCEP STP state information between MCT peers is synchronized using messages that are sent over CCP.
- Only one of the MCT peers send BPDUs towards the MCT Client. It is decided by whosoever is the designated bridge on the ICL.
- New STP States:
  - BLK_BY_ICL state indicates that the superior BPDUs were being received on this interface which could have led to BLOCKING of ICL interface, due to which ICL port guard mechanism has been triggered on this port.
  - FWD_BY_MCT state indicates that the MCT peer has set the CCEP state to forwarding.
  - BLK_BY_MCT state indicates that the MCT peer has set the CCEP state to blocking.

MCT L3 protocols

When configuring dynamic L3 protocol support over MCT, you should consider the following.

- L3 Protocols IS-ISv4, IS-ISv6, BGP4, BGPv6, OSPFv2, OSPFv3, RIP, and RIPng are supported.
- CCEP and CEP are on different L3 networks.
- Any of the L3 protocol can be configured on the CCEP ports and CEP ports.
- An ICL interface can not be configured with L3 routing protocols.
- A CEP Network is reachable via MCT nodes even when the CCEP is down on one of the peers , the traffic will take redundant path via MCT.
- Active and Passive L3 routing protocols interfaces are supported.

MCT feature interaction

Use the following feature matrix when configuring MCT:

TABLE 73 MCT feature interaction matrix

Supported	Not Supported
LACP on Inter-Chassis Link (ICL) and Customer Client Edge Ports (CCEP).	MSTP, VSRP, and PIM.
VRRP on CCEP.	**ESI VLANs** on CCEP or ICL ports.
MRP and **MRP II** supported with the restriction that ICL port cannot be the secondary port of the MRP ring.	**GRE** is not supported on the ICL VE interfaces.
BGP, **IS-IS**, and **OSPF** on CCEP.	**BFD** on CCEP.
802.1ad on CCEP or ICL ports.	**DAI** on CCEP.

TABLE 73 MCT feature interaction matrix (continued)

Supported	Not Supported
Flooding features (VLAN CPU protection, multicast flooding, 802.1ad and others) on cluster VLANs.	**802.1ah** on CCEP or ICL ports.
Multi-VRF	**MSTP**
UDLD as independent boxes.	**VPLS** on ICL ports.
VPLS and **VLL** on CCEP.	**VLL** on ICL ports.
Link OAM as independent boxes.	**MPLS** on CCEP or ICL ports.
802.1ag as independent boxes.	**Hitless Upgrade** is not supported, on the Brocade NetIron MLX Series, and NetIron XMR, however it is compatible. If the operation is performed with cluster configuration the TCP session is reestablished. The MAC addresses from the cluster peers will be re-validated and programmed accordingly. Brocade recommends shutting down all the CCEP on the cluster node so that there is graceful failover and then hitless operation can be performed.
ARP as independent boxes.	**Hitless Failover** is not supported on the Brocade NetIron MLX Series, and NetIron XMR, however it is compatible. If the operation is performed with cluster configuration the TCP session is reestablished. The MACs from the cluster peers will be re-validated and programmed accordingly. Brocade recommends shutting down all the CCEP on the cluster node so that there is graceful failover and then hitless operation can be performed.
STP and RSTP	**Multi-port ARP** will not be allowed on CCEP or ICL ports on the Brocade NetIron MLX Series and NetIron XMR.
Port MAC Security on the node where it is programmed.	**Multiport MAC** is not supported on CCEP or ICL ports. Configuration will be rejected when trying to configure multiport MAC addresses with a port mask which contains either a CCEP or ICL port and vice-versa on the Brocade MLXe, NetIron MLX, and NetIron XMR routers.
802.1x on the node where it is programmed.	
Static MAC configuration - Static MACs are programmed on both local and remote peers as static entries.	

Configure MCT

Active-Active MCT configuration considerations

- On Customer Client Edge Ports (CCEP), MCT does not support 802.1ah (listed in MCT feature interaction on page 565 as unsupported).
- ICL ports should be not be an untagged member of any VLAN. An ICL is preferably a LAG that provides port level redundancy and higher bandwidth for cluster communication.
- On the Brocade NetIron MLX Series and NetIron XMR, ICL ports can be part of MCT VLANs as well as regular VLANs.
- The Brocade NetIron MLX Series, and NetIron XMR devices will disable MAC learning on ICL ports for the VLANs configured in the cluster. However, MAC learning is enabled on ICL port for non-cluster VLANs.
- MAC Database Update Protocol (MUDP) will synchronize all MAC entries for VLANs served by ICL link.
- Cluster ID should be same on both cluster switches.
- Cluster RBridge ID should not conflict with any client RBridge ID or the peer RBridge ID.
- Client RBridge ID is unique and it should be same on the cluster switches.
- You can add any ports to the session VLAN (For the purpose of adding a port to a LAG), but Brocade recommends keeping only ICL ports as tagged members for the session VLAN during operation.

- MCT clients may support 16 members per LAG.
- An ICL interface cannot be configured as the CCEP port in any client.
- CCEP ports on MCT node can be single port or LAG.
- If ICL or client interfaces need to be configured as LAG interface then only the primary port of the LAG needs to be specified in the ICL or client configuration.
- Once the cluster is deployed, only the cluster member VLANs and client isolation mode can be modified. Other configurations are not allowed to change.
- Once the client is deployed, any configuration under client is not allowed to change.
- Clients can be added or deleted even if the cluster is deployed.
- When the cluster is not deployed, then all the clients in the cluster become inactive.
- As soon as a port is configured as an ICL port it is removed from default VLAN.
- If an ICL or CCEP is a LAG interface, the LAG has to be configured separately on each node.

Configuring Active-Active MCT

The following basic steps are required to build an MCT scenario. Refer to Single level MCT example on page 570 for detailed instructions.

1. Create and deploy a LAG that will be used as ICL port.
2. Create and deploy the LAGs facing the clients.
3. Enable Layer 2 switching (**no route-only** command) either globally or on specific interfaces.
4. Create and add ports to a client or member VLAN that they will be using for communication.
   a) At the CCEP or CCP, these ports may be tagged or untagged into the VLAN.
   b) At the ICL, these ports may only be tagged into the VLAN. The ICL ports cannot be untagged in any VLAN.
5. Create a dedicated session VLAN for the ICL interfaces for CCP communication. ICL ports will be tagged into the session VLAN. It is recommended to use a high VLAN number that will not be touched by data VLANs.
6. Create a virtual routing interface and associate this with the session VLAN. This will be used to address the link between the MCT peers.
7. Configure the MCT cluster.
   a) Create a cluster with any name but with a cluster ID matching the MCT peer.
   b) Configure a unique RBridge ID for this peer. The RBridge ID must be unique across all MCT peers and CCEPs.
   c) Configure the session VLAN for the cluster.
   d) Configure one or more client or member VLANs for the cluster.
   e) Configure the ICL port or LAG being used for the cluster.
   f) Configure the MCT peer for this cluster.
   g) Configure the time delay for the LACP blocked state to enable the MCT nodes to process the remote CCEP events. This configuration step is supported on Brocade NetIron MLX and XMR devices only.
8. Deploy the cluster. All attributes except for client or member VLANs cannot be changed after the cluster is deployed.
9. Configure the MCT cluster client instances.

Active-Passive MCT

Active-Passive MCT configuration considerations

- To enable MCT Active-Passive mode, both the MCT nodes must be configured as Active-Passive. If only one node is configured, then cluster CCP session will not come up.
- Active-Passive mode is supported for the following combinations.
  - L2VPN MCT configured.
  - L2 + L2VPN MCT configured.
- Active-Passive mode does not support L2 MCT.
- VLL and VPLS are supported over L2VPN MCT.
- Node where the Pseudo Wire's are Active must be configured as the MCT Active node .
- The CCEP ports must be part of dynamic LAG. Active-Passive mode is not supported on static LAG CCEP ports and single port CCEP.
- CCEP ports elected under passive node will be made LACP-BLOCKED.
- VLL PW's role is based on the client role, the node under which the client ports are active the PW's will be active from that particular node towards the upstream.
- When using VPLS, the Pseudo Wire's role and the client role are independent, and is based on the client role election based on the configuration.

Configuring Active-Passive MCT

After configuring the basic steps required to build an MCT scenario, complete the following steps to configure the MCT cluster.

1. Create a cluster with any name but with a cluster ID matching the MCT peer.
2. Configure a unique RBridge ID for this peer. The RBridge ID must be unique across all MCT peers and CCEPs.
3. Configure the session VLAN for the cluster.
4. Configure one or more client or member VLANs for the cluster.
5. Configure the LAG being used for the cluster.
6. Enable the Active or Passive mode.
7. Configure the client role.
8. Configure the client role revertible mode.
9. Configure the client role revertible timer.
10. Configure the MCT peer for this cluster.
11. Configure the ICL port or LAG being used for the cluster.
12. Configure the MCT peer for this cluster.
13. Configure the time delay for the LACP blocked state to enable the MCT nodes to process the remote CCEP events. This configuration step is supported on Brocade NetIron MLX and XMR devices only.
14. Deploy the cluster. All attributes except for client or member VLANs cannot be changed after the cluster is deployed.
15. Configure the MCT cluster client instances.

    After deploying the cluster, both cluster peers will begin exchanging their cluster mode and client role information. The selection will take place based on the criteria listed in Table 74.

TABLE 74 Client role election criteria

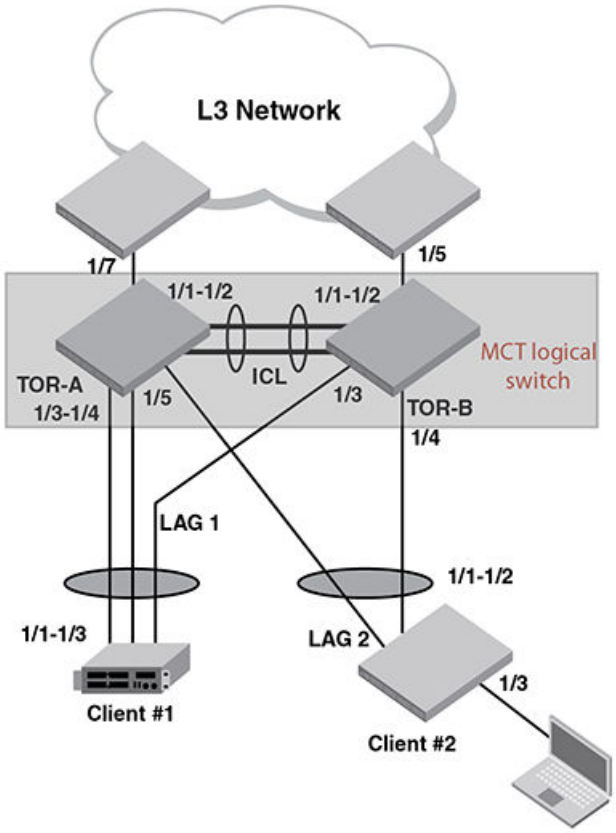
PE1 client role configuration	PE2 client role configuration	Active links of MCT
Active	Active	Based on the RBridge ID
Passive	Active	PE2
N/A	Active	PE2
Active	Passive	PE1
Passive	Passive	Based on the RBridge ID
N/A	Passive	PE1
Active	N/A	PE1
Passive	N/A	PE2
N/A	N/A	Based on the RBridge ID

Sample Active-Passive MCT cluster configurations

```
Cluster MCT1 1
  rbridge-id 100
  mode-active-passive
  client-role active
  client-role-revertible-delay timer 5
  l2vpn-peer 12.12.12.12 rbridge-id 101
  deploy
 client c1
  rbridge-id 100
  client-interface ethernet 1/6
  deploy
Cluster MCT2 1
  rbridge-id 200
  mode-active-passive
  client-role-revertible-delay timer 5
  l2vpn-peer 11.11.11.11 rbridge-id 101
  client-interfaces sync_ccep_early lacp-delay 5
        deploy
 client c1
  client-role passive
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
```

Single level MCT example

FIGURE 164 Single level MCT



The following steps are task for configuring a MCT scenario as shown in **Figure 176** .

NOTE
Save the current configuration files for both chassis operating in standalone mode before you begin creating a MCT.

TOR-A (Top of rack MCT capable switch)

See **Figure 176 .**

Creating LAG-1

1. You can either assign a LAG **ID** explicitly or it will be automatically generated by the system. The LAG **ID** stays the same across system reload and hitless upgrade.

   The command to configure LAGs allows explicit configuration of the LAG ID for static and dynamic LAGs.

To create a LAG with the LAG ID option, enter a command such as the following.

```
device(config)# lag 1 dynamic
device(config-lag-1)#
```

Syntax: [no] lag name [**static** | **dynamic**] [**id number**]

The **ID** parameter is optional. The value of the **ID** parameter that you can enter is from 1 to 256. If you do not enter a LAG **ID** , the system will generate one automatically. Once the LAG **ID** is generated the system will save it in the configuration file along with the LAG name, therefore the value will stay the same across system reload.

> **NOTE**
> The LAG ID parameter is for static and dynamic LAGs only. No explicit configuration of a LAG id is allowed on keepalive LAGs.

The **static** parameter specifies that the LAG with the name specified by the *lag-name* variable will be configured as a static LAG.

The **dynamic** option specifies that the LAG with the name specified by the *lag-name* variable will be configured as a dynamic LAG.

2. Define the ports the LAG will be using as shown in the following.

```
device(config-lag-1)# ports ethernet 1/1 to 1/2
```

Syntax: [no] ports ethernet [**slot/port**] | **to** | [**slot/port**]

Use the appropriate *slot/port* variable to specify a Ethernet port within the LAG that you want to enable.

3. The primary port must be explicitly assigned using the **primary-port** command. To designate the primary port for the static LAG "1", use the following command.

```
device(config-lag-1)# primary-port 1/1
```

Syntax: [no] primary-port slot/port

Once a primary port has been configured for a LAG, all configurations that apply to the primary port are applied to the other ports in the LAG.

> **NOTE**
> This configuration is only applicable for configuration of a static or dynamic LAGs

4. After configuring a LAG, you must explicitly enable it before it begins aggregating traffic. This task is accomplished by executing the deploy command within the LAG configuration. After the deploy command runs, the LAG is in the aggregating mode. Only the primary port within the LAG is available at the individual interface level. Any configuration performed on the primary port applies to all ports within the LAG. The running configuration will no longer display deployed LAG ports other than the primary port.

   To deploy a LAG, at least one port must be in the LAG and the primary port must be specified for non keep-alive LAGs. After a non keep-alive LAG is deployed, a trunk is formed. If there is only one port in the LAG, a single port is formed. For a dynamic LAG, LACP is stared for each LAG port.

   Use a command such as the following to deploy LAG 1

```
device(config-lag-1)# deploy
```

Syntax: [no] deploy [**forced** | **passive**]

When the **deploy** command is executed:

- – For a static and dynamic LAGs, the current veto mechanism is invoked to make sure the LAG can be formed. If the LAG is not vetoed, a **no** is formed with all the ports in the LAG.
  – For dynamic LAGs, LACP is activated on all LAG ports. When activating LACP, use active mode if passive is not specified; otherwise, use passive mode.
  – For a keep-alive LAGs, a LAG is formed, and LACP is started on the LAG port.

  Once the deploy command is issued, all LAG ports will behave like a single port.

  If the **no deploy** command is executed, then the LAG is removed. For dynamic LAGs, LACP is de-activated on all of the LAG ports.

  If the **no deploy** command is issued and more than 1 LAG port is not disabled the command is aborted and the following error message is displayed: "Error 2 or more ports in the LAG are not disabled, un-deploy this LAG may form a loop - aborted." Using the **forced** keyword with the **no deploy** command in the previous situation, the un-deployment of the LAG is executed.

5. Assign a name to an individual port within a LAG using the **port-name** command within the LAG configuration as shown in the following. Using the **port-name** command is optional.

```
device(config-lag-1)# port-name ICL-to-TOR-B:1/1 ethernet 1/1
device(config-lag-1)# port-name ICL-to-TOR-B:1/2 ethernet 1/2
```

   Syntax: [no] port-name text ethernet [**slot/port**] | **pos** [**slot/port**]

   The *text* variable specifies the port name. The name can be up to 50 characters long.

   Use the **ethernet** option with the appropriate *slot/port* variable to apply the specified name to an Ethernet port within the LAG.

   Use the **pos** option with the appropriate *slot/port* variable to apply the specified name to a Packet-over-SONET port within the LAG.

Creating LAG 2

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create LAG 2 as shown below.

```
device(config)# lag 2 dynamic id 2
device(config-lag-2)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-2)# ports ethernet 1/3  to 1/4
```

3. Deploy the LAG 2 as shown below.

```
device(config-lag-2)# deploy
```

4. Assign a name to an individual port within a LAG.

```
device(config-lag-2)# port-name lag-client-1:1/1 ethernet 1/3
device(config-lag-2)# port-name lag-client-1:1/2 ethernet 1/4
```

Creating LAG 3

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create LAG 3 as shown below.

```
device(config)# lag 3 dynamic id 3
device(config-lag-3)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-3)# ports ethernet 1/5
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-3)# primary-port 1/5
```

4. Deploy the LAG 3 as shown below.

```
device(config-lag-3)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-2)# port-name lag-client-2:1/1 ethernet 1/5
device(config-lag-2)# port-name lag-client-1:1/2 ethernet 1/4
```

Enable layer 2 switching

By default, Brocade devices support routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command. The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. To enable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and add the **no route-only** command.

NOTE
On the Brocade NetIron XMR Series and Brocade NetIron MLX Series routers, **route-only** is the default condition. Because **route-only** is the default condition, it will not be displayed in the configuration. If you use **no route-only** to enable switching, the **no route-only** command will be displayed in the configuration.

NOTE
On the Brocade NetIron CES Series and Brocade NetIron CER Series devices, **route-only** is disabled by default. Therefore, if **route-only** is enabled on a Brocade NetIron CES Series or Brocade NetIron CER Series device, it will be displayed in the configuration.

Use commands such as the following to enable Layer 2 switching.

```
device(config)# no route-only
```

Syntax: [no] route-only

Creating VLANs for client traffic

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating VLANs.

Creating VLANs for client traffic

1. At the global CONFIG level assign an ID to the VLAN.

```
device(config)# vlan 2
```

2. Add the client ports to the VLAN as either tagged or untagged. In this example the client ports are untagged.

```
device(config-vlan-2)# untag eth 1/3 to 1/5
```

3. Add the ICL port or LAG to the VLAN as tagged. ICL ports cannot be untagged in any VLAN and will automatically be removed from the default VLAN upon MCT cluster configuration.

```
device(config-vlan-2)#tag eth 1/1 to 1/2
```

Create the session VLAN

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating VLANs.

1. At the global CONFIG level assign an ID to the VLAN .

```
device(config)# vlan 4090 name Session-VLAN
```

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-4090)# tagged ether 1/1 to 1/2
```

3. Configure a virtual routing interface on each IP protocol VLAN, then configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
device(config-vlan-4090)# tagged ether 1/1 to 1/2
device(
config-vlan-4090)# router-interface ve 100
```

Assign the hostname (optional)

To configure a system name, enter commands such as the following.

```
device(config)# hostname TOR-A
```

Enabling interfaces

The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the ports on a device for ethernet interfaces 1/1, 1/3, and 1/5, enter the following commands.

```
device(config)# interface ether 1/1
device(config-if-e10000-1/1)# enable
device(config)# interface ether 1/3
device(config-if-e10000-1/3)# enable
device(config)# interface ether 1/5
device(config-if-e10000-1/5)# enable
```

Syntax: [no] enable

Assigning a port name (optional)

A port name can be assigned to help identify interfaces on the network. You can assign a port name to physical ports, virtual routing interfaces, and loopback interfaces. To assign a name to a ports 1/6 and 1/7, enter the following commands.

```
device(config)# interface ethe 1/6
device(config-if-e10000-1/6)# port-name CEP-PC
device(config-if-e10000-1/6)# enable
device(config)# interface ethe 1/7
device(config-if-e10000-1/7)# port-name to-L3-ECMP
device(config-if-e10000-1/7)# enable
```

Syntax: [no] port-name *text*

Syntax: [no] enable

The *text* parameter is an alphanumeric string. The name can have up to 255 characters on a device and can include blanks. You do not need to use quotation marks around the string, even when it contains blanks.

Adding a virtual interface

Add a virtual interface and configure an IP address on the interface by entering commands such as the following.

```
device(config)# interface ve 100
device(config-vif-100)# ip address 1.1.1.1/24
```

Syntax: [no] interface [**ve** *ve-id*]

Syntax: [no] ip address *ip-addrip-mask*

The *ve-id* variable allows you to specify a VE interface ID.

Configuring the cluster operation mode

See **Figure 176** and Creating LAG-1 on page 570.

1. To configure a device with cluster ID 1, enter a command such as the following.

   ```
   device(config)# cluster TOR 1
   ```

   Syntax: [no] cluster *cluster-namecluster-id*

   The *cluster-name* parameters specify the cluster name with a limit of 64 characters.

   The *cluster-id* parameters specify the cluster ID (1-65535).

2. Configure the local RBridge ID for the cluster. This RBridge ID is used by the peer to communicate with this cluster node and to define CCEPs. The RBridge ID needs to be unique across the cluster and unique between MCT peer Bridge IDs as well as cluster client instances. To configure the local rbridge, enter a command such as following

   ```
   device(config-cluster-TOR)#rbridge-id 1
   ```

   Syntax: [no] rbridge-id *id*

   The *id* parameter specifies the remote bridge ID. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN can be in the range 1-4090, but it cannot be the default VLAN. A check is made during the cluster deploy in addition to a dynamic check. The default VLAN cannot be changed to a VLAN which is already defined as cluster session. Note: ICL ports must be tagged within the session VLAN. Enter a command such as the following to create the session VLAN.

   ```
   device(config-cluster-TOR)# session-vlan 4090
   ```

   Syntax: [no] session-vlan *vlan-id*

   The *vlan-id* parameter specifies the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

   ```
   device(config-cluster-TOR)# member-vlan
    2
   ```

   Syntax: [no] member-vlan *x* [**to** *y*]

   > **NOTE**
   > The VLAN range is allowed to change even if cluster is deployed.

5. Specify the ICL for the cluster. The ICL interface can be a single link or trunk port. If it is a trunk port, it should be the primary port of the trunk . Only one ICL is supported. Enter a command such as the following to create the ICL for the cluster.

```
device(config-cluster-TOR)#icl
 TOR ethernet 1/1
```

Syntax: [no] icl *icl-name* **ethernet** *x/y*

The *icl-name* parameter can be up to 64 characters in length.

The **ethernet** *x/y* parameter is the ICL interface.

6. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
device(config-cluster-TOR)# peer 1.1.1.2 rbridge-id 2 icl TOR
```

Syntax: [no] peer *peer-ip* **rbridge-id** *peer-rbridge* **icl** *map-icl*

The *peer-ip* parameter should be in same subnet as that of cluster management interface.

The *peer-rbridge* parameter should be different from cluster rbridge and any other client in the cluster

The *map-icl* parameter is the ICL name to reach this cluster peer.

7. The cluster can be deployed separately without any clients configured. The **deploy** command brings the cluster into effect. The following can be changed when the cluster is deployed:
   - Client isolation mode
   - Member VLANs
   - Clients added and removed.
8. The **deploy** command also preforms a consistency check of the entire cluster configuration. If anything is amiss, an error message is sent. The following specific information is checked during deployment:
   - If the cluster management VLAN is configured
   - If the cluster peer is configured
   - If the cluster ICL is configured

   Enter a command such as the following to deploy the cluster configuration.

```
device(config-cluster-TOR)# deploy
```

Syntax: [no] deploy

Creating cluster client 1

See **Figure 176** .

1. Create a cluster client instance and change the mode to the client instance. If an instance is already present, then directly change the mode to the client instance mode.

```
device(config-cluster-TOR)# client client-1
```

Syntax: [no] client *client-name*

The *client-name* parameter can be 64 characters (maximum).

If it is a two port MCT, the maximum clients supported on the Brocade NetIron XMR Series or Brocade NetIron MLX Series system is 1536/2.

If it is a two port MCT, the maximum clients supported on the NetIron CES or NetIron CER system is 1535/2.

2. Configure the local RBridge ID for the cluster. This RBridge ID is used by the peer to communicate with this cluster node. To configure the local rbridge, enter a command such as following

```
device(config-cluster-TOR-client-1)#rbridge-id 100
```

Syntax: [no] rbridge-id *id*

The *id* parameter specifies the local bridge id. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN is the VLAN used by the cluster for control operations. Add the (CCEP or CEP) interfaces to the cluster client instance. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG.

```
device(config-cluster-TOR-client-1)#client-interface ether 1/3
```

Syntax: [no] client-interface *interface* **interface : ethernet** *x/y*

The **ethernet***x/y* parameter is the ethernet interface.

4. Deploy the cluster client. If cluster is not deployed, the configuration will be taken but the client state machine will not be started. The consistency checks for client will be done at the time of client deploy. The following configuration checks will be preformed:

- – Client interface is configured
  – Client interface is not same as any other client interface or ICL interface
  – Client RBridge ID is not same as cluster rbridge or any peer rbridge

Once the client is deployed, the configuration inside the client will not be allowed to change, To deploy the client configuration, enter a command such as the following.

```
device(config-cluster-TOR-client-1)deploy
```

Syntax: [no] deploy

Create cluster client 2

See **Figure 176** and Create cluster client 1 on page 583 for additional information for creating cluster clients.

1. Create a cluster client instance.

```
device(config-cluster-TOR)# client client-2
```

2. Configure the client RBridge ID.

```
device(config-cluster-TOR-client-2)#rbridge-id 200
```

3. Create a cluster client interface.

```
device(config-cluster-TOR-client-2)#client-interface ether 1/5
```

4. Deploy the cluster client.

```
device(config-cluster-TOR-client-2)deploy
```

TOR-B

Creating LAG-1

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create a LAG with the LAG ID option.

```
device(config)# lag 1 dynamic id 1
device(config-lag-1)#
```

2. Define the port the LAG will be using:

```
device(config-lag-1)# ports ethernet 1/6
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-1)# primary-port 1/1
```

4. Deploy a LAG as shown below.

```
device(config-lag-1)# deploy
```

Assign a name to an individual port within a LAG.

```
device(config-lag-1)# port-name lag-client-2:1/2 ethernet 1/6
```

Creating LAG 2

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
device(config)# lag 2 dynamic id 2
device(config-lag-2)#
```

2. Define the port the LAG will be using.

```
device(config-lag-2)# ports ethernet 1/7
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-2)# primary-port 1/7
```

4. Deploy a LAG as shown below.

```
device(config-lag-2)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-2)# port-name lag-client-3:1/2 ethernet 1/7
```

Creating LAG 3

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
device(config)# lag 3 dynamic id 3
device(config-lag-3)#
```

2. Define the port the LAG will be using.

```
device(config-lag-3)# ports ethernet 1/3 to 1/4
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-3)# primary-port 1/3
```

4. Deploy a LAG as shown below.

```
device(config-lag-3)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-3)# ICL-to-TOR-A:1/3 ethernet 1/3
device(config-lag-3)# ICL-to-TOR-A:1/4 ethernet 1/4
```

Creating LAG 4

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create a LAG as shown below.

```
device(config)# lag 4 dynamic id 4
device(config-lag-4)#
```

2. Define the port the LAG will be using.

```
device(config-lag-4)# ports ethernet 1/5
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-4)# primary-port 1/5
```

4. Deploy a LAG as shown below.

```
device(config-lag-4)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-3)# lag-client-1:1/2 ethernet 1/5
```

Enable layer 2 switching

By default, Brocade devices support routing over layer 2 switching. You can enable layer 2 switching globally or on individual port using the **no route-only** command. The **no route-only** and **route-only** commands prompts you for whether or not you want to change the "route-only" behavior. You must enter **y** if you want to proceed or **n** if you do not. To enable Layer 2 switching only on a specific interface, go to the interface configuration level for that interface, and add the **no route-only** command.

NOTE

On the Brocade NetIron CES/CER, the **no route-only** interface configuration is only valid when the interface is strictly untagged in a regular VLAN. Configure **no route-only** at the global configuration if tagged/dual-mode ports are being used in the system.

NOTE

On the Brocade NetIron XMR Series and Brocade NetIron MLX Series devices, **route-only** is the default condition. Because **route-only** is the default condition, it will not be displayed in the configuration. If you use **no route-only** to enable switching, the **no route-only** command will be displayed in the configuration.

NOTE
On the Brocade NetIron CES Series and Brocade NetIron CER Series devices, **route-only** is disabled by default. Therefore, if **route-only** is enabled on a Brocade NetIron CES Series or Brocade NetIron CER Series device, it will be displayed in the configuration.

Use the following command to enable Layer 2 switching.

```
device(config)# no route-only
```

Syntax: [no] route-only

The following warning messages are displayed by the system when there is a conflict between global and interface level route only configurations on a VLAN tagged interface.

Warning message	Configuration
`"no route-only" interface configuration conflicts with the global "route-only" configuration, This configuration will be applied on 1/5 interface when the interface becomes strictly untagged interface`	Global config : **route-only** Interface config on vlan tagged port: **no route-only**
`"route-only" interface configuration conflicts with the global "no route-only" configuration, This configuration will be applied on 1/6 interface when the interface becomes strictly untagged interface`	Global config: **no route-only** Interface config on vlan tagged port: **route-only**

Creating VLANs

See Single level MCT example on page 570

VLAN 1

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN.

```
device(config)# vlan 2
```

2. Add ports to that VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-2)# tagged e 1/1 to 1/8
device(config-vlan-2)# no untag ether 1/3 to 1/4
```

VLAN 2

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN 2.

```
device(config)# vlan 2 client-vlan
```

2. Add ports to that VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-2)# untag ether 1/5 to 1/7
device(config-vlan-2)# tagged ether 1/3 to 1/4
```

Create the session VLAN

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. At the global CONFIG level assign an ID to the VLAN 4090.

```
device(config)# vlan 4090 name Session-VLAN
```

Syntax: [no] vlan vlan-id name *vlan-name*

VLAN IDs can be in the range of 1 - 4090. Use the **no** form of the command to delete the VLAN from the configuration.

The VLAN ID range above 4090 has been reserved for current and future features for internal control purposes.

In addition to a VLAN number, you can assign a name to a VLAN by entering name *vlan-name*. Enter up to 32 characters for name.

2. Add ports to the VLAN and specify if the ports are tagged or untagged.

```
device(config-vlan-4090)# tagged ether 1/3 to 1/4
```

3. Configure the appropriate IP routing parameters on each of the virtual routing interfaces.

```
device(config-vlan-4090)# tagged ether 1/3 to 1/4
device(
config-vlan-4090)# router-interface ve 100
```

Assign the hostname - optional

Configure a system name for the device and save the information locally in the configuration file for future reference. The information is not required for system operation but recommended. When you configure a system name, it replaces the default system name in the CLI command prompt.

To configure a system name, enter a command such as the following.

```
device(config)# hostname TOR-B
```

Syntax: [no] hostname name

The *name* can be up to 255 alphanumeric characters. The text strings can contain blanks.

Enabling interfaces

The ports can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the ports on a device for ethernet interfaces 1/3, 1/5, 1/6,and 1/7, enter the following commands.

```
device(config)# interface ether 1/3
device(config-if-e10000-1/3)# enable
device(config)# interface ether 1/5
device(config-if-e10000-1/5)# enable
device(config)# interface ether 1/6
device(config-if-e10000-1/6)# enable
device(config)# interface ether 1/7
device(config-if-e10000-1/7)# enable
```

Syntax: [no] enable

Adding a virtual interface

Add a virtual interface and configure an IP address on the interface by entering commands such as the following.

```
device(config)# interface ve 100
device(config-vif-100)# ip address 1.1.1.2/24
```

Syntax: [no] interface [**ve** *ve-id*]

Syntax: [no] ip address *ip-addrip-mask*

The *ve-id* variable allows you to specify a VE interface ID.

Configuring the cluster operation mode

The cluster can be deployed separately without any client configured. When the cluster is deployed, it will check all the deployed clients and start the state machine for the clients. See Single level MCT example on page 570.

1. Configure one cluster ID or name on the device so that all route-reflector clients for the device become members of the cluster. To configure a device with cluster ID 1, enter the following command.

   ```
   device(config)# cluster TOR 1
   ```

   Syntax: [no] cluster *cluster-namecluster-id*

   The *cluster-name* parameters specify the cluster name with a limit of 64 characters.

   The *cluster-id* parameters specify the cluster ID (1-65535). The default is the device ID.

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

   ```
   device(config-cluster-TOR)#rbridge-id 2
   ```

   Syntax: [no] rbridge-id *id*

   The *id* parameters specify the remote bridge ID. Possible values are 1 - 35535 (16 bit value).

3. The cluster session VLAN is in range 1-4090 but cannot be default VLAN. A check is made during the cluster deploy and in addition to a dynamic check. The default VLAN cannot be changed to a VLAN which is already defined as cluster session

   ```
   device(config-cluster-TOR)# session-vlan
    4090
   ```

   Syntax: [no] session-vlan *vlan-id*

   The *vlan-id* parameters specify the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

   ```
   device(config-cluster-TOR)# member-vlan
    2
   ```

   Syntax: [no] member-vlan x to *y*

   > **NOTE**
   > The VLAN range is allowed to change even if cluster is deployed.

   The new VLAN range will over-ride the previous configured range.

5. Specify the ICL for the cluster.The ICL interface can be a single link or trunk port. If it is a trunk port, it should be the primary port of the trunk. Only one ICL is supported.

   ```
   device(config-cluster-TOR)#icl
    TOR ethernet 1/3
   ```

   Syntax: [no] icl *icl-name* **ethernet** *x/y*

   The *icl-name* parameter can be 64 characters (maximum).

The **ethernet***x/y* parameter is the ICL interface.

6. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
device(config-cluster-TOR)# peer 1.1.1.1 rbridge-id 1 icl TOR
```

Syntax: [no] peer *peer-ip* **rbridge-id** *peer-rbridge* **icl** *map-icl*

The *peer-ip* parameter should be in same subnet as that of cluster management interface.

The *peer-rbridge* parameter should be different from cluster rbridge and any other client in the cluster

The *map-icl* parameter is the ICL name to reach this cluster peer.

7. Clusters can be deployed separately without any client configured. The **deploy** command brings the cluster into effect. Once the cluster is deployed, the configuration inside the cluster can not be changed. The **deploy** command also preforms a consistency check of the entire cluster configuration. If anything is amiss, an error message is sent.The specific information checked during deploy:
   - If the cluster management VLAN is configured
   - If the cluster peer is configured
   - If the cluster ICL is configured

   Enter a command such as the following to deploy the cluster configuration.

```
device(config-cluster-TOR)# deploy
```

Syntax: [no] deploy

Create cluster client 1

1. Create a cluster client instance and change the mode to the client instance. If an instance is already present, then directly change the mode to client instance mode.

```
device(config-cluster-TOR)# client client-1
```

Syntax: [no] client *client-name*

The *client-name* parameter can be 64 characters (maximum).

If it is a two port MCT, the maximum clients supported on the Brocade NetIron XMR Series or Brocade NetIron MLX Series is 1536/2.

If it is a two port MCT, the maximum clients supported on the Brocade NetIron CES Series or Brocade NetIron CER Series system is 50/2.

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

```
device(config-cluster-TOR-client-1)#rbridge-id 100
```

Syntax: [no] rbridge-id *id*

The *id* parameters specify the remote bridge id. Possible values are 1 - 35535 (16 bit value).

3. Create a cluster client interface. The interface can be a single link or trunk port. If it is trunk port, it should be the primary port of the trunk.

```
device(config-cluster-TOR-client-1)#client-interface ether 1/5
```

Syntax: [no] client-interface *interface* **interface : ethernet** *x/y*

The **ethernet***x/y* parameter is the ethernet interface.

4. Deploy the cluster client. If cluster is not deployed, the configuration will be taken but the client FSM will not be started. The consistency checks for client will be done at the time of client deploy. The following configuration checks will be preformed:
   - Client interface is configured
   - Client interface is not same as any other client interface or ICL interface
   - Client RBridge ID is not same as cluster rbridge or any peer rbridge

   Once the client is deployed, the configuration inside the client will not be allowed to change, To deploy the client configuration, enter a command such as the following.

   ```
   device(config-cluster-TOR-client-1)deploy
   ```

   Syntax: [no] deploy

Create cluster client 2

See Single level MCT example on page 570 and Create cluster client 1 on page 583 for additional information on creating cluster clients.

1. Create a cluster client instance and change the mode to the client instance mode.

   ```
   device(config-cluster-TOR)# client client-2
   ```

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

   ```
   device(config-cluster-TOR-client-2)#rbridge-id 200
   ```

3. Create a cluster client interface.

   ```
   device(config-cluster-TOR-client-2)#client-interface ether 1/6
   ```

4. Deploy the cluster client.

   ```
   device(config-cluster-TOR-client-2)deploy
   ```

Create cluster client 3

See Single level MCT example on page 570 and Create cluster client 1 on page 583 for additional information on creating cluster clients.

1. Create a cluster client instance and change the mode to the client instance mode.

   ```
   device(config-cluster-TOR)# client client-3
   ```

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

   ```
   device(config-cluster-TOR-client-3)#rbridge-id 300
   ```

3. Create a cluster client interface. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG.

   ```
   device(config-cluster-TOR-client-2)#client-interface ether 1/7
   ```

4. Deploy the cluster client.

   ```
   device(config-cluster-TOR-client-2)deploy
   ```

Configuring Client-1

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create LAG 1 as shown below.

```
device(config)# lag 1 dynamic id 1
device(config-lag-1)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-1)# enable ethernet 1/1 to 1/3
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-1)# primary-port 1/1
```

4. Deploy the LAG as shown below.

```
device(config-lag-1)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-1)# port-name lag-to-TOR-A ethernet 1/1
device(config-lag-1)# port-name lag-to-TOR-B ethernet 1/3
```

6. The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the port on a device for ethernet interface 1/1, enter the following command.

```
device(config-if-e10000-1/1)# enable
```

Configuring Client 2

See **Figure 176** and Creating LAG-1 on page 570 for additional information on creating a LAG.

1. Create a LAG with the LAG ID option, enter a command such as the following.

```
device(config)# lag 1 dynamic id 1
device(config-lag-1)#
```

2. Define the ports the LAG will be using.

```
device(config-lag-1)# ports ethernet 1/1 to 1/2
```

3. The primary port must be explicitly assigned using the **primary-port** command.

```
device(config-lag-1)# primary-port 1/1
```

4. Deploy the LAG as shown below.

```
device(config-lag-1)# deploy
```

5. Assign a name to an individual port within a LAG.

```
device(config-lag-1)# port-name lag-to-TOR-A ethernet 1/1
device(config-lag-1)# port-name lag-to-TOR-B ethernet 1/2
```

6. The port can be made inactive (disable) or active (enable) by selecting the appropriate status option. The default value for a port is disabled. To enable the port on a device for ethernet interface 1/1 and 1/3, enter the following commands.

```
device(config)# interface ether 1/1
device(config-if-e10000-1/1)# enable
device(config)# interface ether 1/3
device(config-if-e10000-1/3)# enable
```

7. Assign a name to an individual port within a LAG.

```
device(config-if-e10000-1/3)# port-name host-to-PC
device(config-if-e10000-1/3)# enable
```

Optional cluster operation features

A cluster can operate in two modes:

Cluster Failover Mode

Fast-failover (default) - As soon as the ICL interface goes down the CCP goes down. All the remote MACs are flushed.

Slow-failover - Even if the ICL interface goes down the CCP waits for the hold-time before making the CCP down. Remote MACs are flushed only when the CCP is down.

To disable the fast-failover mode, enter a command such as the following.

```
device(config-cluster-TOR)#peer 1.1.1.1 disable-fast-failover
```

Syntax: [no] peer *peer-ip* **disable-fast-failover**

Client isolation mode

NOTE
The CLI will allow modification of the **client-islolation** mode on MCT cluster nodes even when the cluster is deployed. You must create the same isolation mode on both cluster nodes.

The client operates in the following mode:

Strict mode: When the CCP goes down, the client interfaces on both the cluster nodes are administratively shutdown. In this mode, the client is completely isolated from the network if CCP is not operational.

```
device(config-cluster-TOR)#client-isolation strict
```

Syntax: [no] client-isolation strict

Shutdown all client interfaces

Use the **client-interfaces shutdown** command when preforming a hitless-upgrade operation. This command can be used to shutdown all the local client interfaces in the cluster. This would result in failover of traffic to the cluster peer.

```
device(config-cluster-TOR)#client-interfaces shutdown
```

Syntax: [no] client-interfaces shutdown

Client interfaces delay

Use the **client-interfaces delay** command to set the delay before brining up the CCEP port. This command is used to set the delay, so that after a node is reloaded, with just L2vpn peer alone, the delay to bring up the CCEP port will be the designated value.

```
device(config-cluster-TOR)#client-interfaces delay 60
```

Syntax: [no] client-interfaces delay *time insec*

The default value for delay is 90 seconds. The acceptable values range between 20 to 1800 seconds.

Active/Passive mode

To configure MCT to operate in the Active/Passive mode, set the mode for the cluster. The cluster mode should be configured on both the node. If both nodes are not configured Active-Passive, then the CCP will not be brought up. This configuration is not allowed after the cluster is deployed.

```
device(config-cluster-c)# mode-active-passive
```

Syntax: [no] mode-active-passive

Client-role

The **client-role** command is used for configuring the per client role as Active or Passive. When the global and per client role configurations are present, then per client role configuration takes the highest precedence.

If Global and per client role configurations is not present, then the rbridge-id used for client role election, the node with the lowest rbridge-id will be Active.

This global command will be applicable for all the MCT clients.

```
device(config-cluster-c)#client-role active
```

This command will be applicable for a single MCT client.

```
device(config-cluster-c-client-c1)# client-role passive
```

Syntax: [**no**] **client-role** *role*

The *role* parameter specifies if the client role is active or passive.

Client-role-revertible-delay timer

The **client-role-revertible-delay timer** command is used for configuring client role revertible delay mode in case of port flaps, and to revert the client role after client role switch.

```
device(config-cluster-c-client-c1)#client-role-revertible-delay timer 5
```

Syntax: [no] client-role-revertible-delay timer *value*

Enter a value of 1-240 minutes. Default value is 1 minute.

Displaying cluster information

Use the **show cluster** command to display the entire cluster configuration and **show tech cluster** command to display cluster configuration and operation information. See the *Brocade MLXe, NetIron MLX, and NetIron XMR Diagnostic Reference* for additional information on these commands.

Keep-alive VLAN

CCRR message are used to exchange information between peers. When the CCP is up, CCRR messages are sent over CCP. When the CCP client reachability is down, you can use the **keep-alive-vlan** command under cluster context so CCRR messages are periodically

sent over the keep-alive-vlan. Only one VLAN can be configured as a **keep-alive-vlan** . The keep-alive VLAN cannot be a member VLAN of the MCT and this VLAN can be tagged or untagged. A port in keep-alive-vlan cannot be assigned to another VLAN.

```
device(config-cluster-TOR)#keep-alive-vlan 10
```

Syntax: [no] keep-alive-vlan *vlan-id*

The *vlan_id* parameters specify the VLAN range. Possible values are 1 - 4090.

When the CCP is down:

- If **keep-alive-vlan** is configured, then CCRR messages are sent periodically for every 1 second over that VLAN.
- When CCP is down and keep-alive vlan is configured, Master/Slave selection is based on following criteria:

1. If one node's CCEPs are up and other node's CCEPs are down then the node with local CCEPs down becomes Slave
2. Otherwise, the node with higher RBridge ID becomes Slave.
    - If no packetsare received from the peer for a period of 3 seconds, then the peer box is considered down.
    - If **keep-alive-vlan** is not configured and both the peers are up, then both peers keep forwarding the traffic independently.

Keep-alive timers and hold-time

To specify the **keep-alive timers** and **hold-time** for the peers, enter a command such as the following.

```
device(config-cluster-TOR)# peer 1.1.1.1 timers keep-alive 40 hold-time 120
```

Syntax: [no] peer *peer-ip* **timers keep-alive** *keep-alivetime* **hold-time** *hold-time*

The *peer-ip* parameter should be in same subnet as that of cluster management interface.

The *keep-alive time* parameter can be 0 to 21845 (default 30 seconds.)

The *hold-time* parameter can be 3 to 65535 (default 90 seconds) must be at least 3 times the keep alive time.

> NOTE
> Keep-alive-vlan and keep-alive timers are not related. The keep-alive timer is used by CCP.

L2 protocol forwarding

MCT will forward or drop L2 protocol packets when corresponding features are disabled. The packets will either be forwarded as regular multicast that floods to the VLAN or dropped. When forwarded, the packet received from ICL will not be forwarded to CCEP port if the peer MCT switch has the reachability to the same cluster client.

When designing a network, the ICL port or LAG must have enough bandwidth to support all the traffic from clients (in case of client links connected to one node failure case).

For L2 forwarding, appropriate CAM profiles need to be used to be able to program all the MAC entries into the CAM (especially when using LAG interfaces on MCT nodes for ICL and client interfaces).

By default , MCT acts as a hub for STP, or RSTP. Switches connected to MCT can run STP normally. When STP, RSTP, or MSTP is enabled, the L2 protocol forwarding configuration is ignored and has no effect.

To configure L2 protocol forwarding globally, enter a command such as the following.

```
device(config)#cluster-l2protocol-forward
```

To disable L2 protocol forwarding on an interface, enter a command such as the following.

```
device(config-if-e1000-1/2)#cluster-l2protocol-foward disable
```

To remove L2 protocol forwarding configuration on an interface, enter a command such as the following

```
device(config-if-e1000-1/2)#no cluster-l2protocol-foward <enable | disable>
```

Syntax: [no]cluster-l2protocol-forward [**enable** | **disable**]

Interface level configuration overwrites the global level configuration.

TABLE 75 L2 protocol forwarding action -MCT switch and non - MCT switch

Protocol	Destination MAC	Non- MCT switch forwarding action	MCT switch forwarding action
Untagged 802.1Q BPDU	01-80-c2-00-00-00	Flood to the VLAN	Forward to ports that are enabled by the **cluster-l2protocol-forward** command.
Tagged 802.1Q BPDU	01-80-c2-00-00-00	Flood to the VLAN	Forward to ports that are enabled by the **cluster-l2protocol-forward** command.
802.1Q Provider BPDU	01-80-c2-00-00-08	Dropped on NetIron CESand NetIron CER. Flood to the VLAN on NetIron XMR and NetIron MLX	Same as non-MCT switch
802.3 Slow Protocols (e.g. LACP)	01-80-c2-00-00-02	Dropped	Same as non-MCT switch
802.1X PAE address	01-80-c2-00-00-03	Dropped	Same as non-MCT switch
802.1Q Provider Bridge GVRP	01-80-c2-00-00-0D	Flood to the VLAN	Same as non-MCT switch
802.1AB LLDP	01-80-c2-00-00-0E	Flood to the VLAN	Same as non-MCT switch
802.1D GMRP	01-80-c2-00-00-20	Flood to the VLAN	Same as non-MCT switch
802.1Q GVRP	01-80-c2-00-00-21	Flood to the VLAN	Same as non-MCT switch
Foundry MRP (Metro Ring Protocol)	03-04-80-00-00-00	Flood to the VLAN	Same as non-MCT switch
Foundry FDP (Foundry Discovery Protocol)	01-e0-52-cc-cc-cc	Flood to the VLAN	Same as non-MCT switch
CDP	01-00-00-cc-cc-cc	Flood to the VLAN	Same as non-MCT switch
PVST	01-00-0c-cc-cc-cd	Flood to the VLAN	Same as non-MCT switch
SuperSpan	03-80-c2-xx-xx-00	Flood to the VLAN	Same as non-MCT switch
VSRP Control	03-04-80-00-01-00	Flood to the VLAN	Same as non-MCT switch
VSRP Source	03-04-80-00-01-01	Flood to the VLAN	Same as non-MCT switch
Loop detection MAC	Base MAC address | 0x03000000	Flood to the VLAN	Same as non-MCT switch

Port loop detection

Port loop detection is used to detect L2 loops in MCT (due to misconfiguration). When using MCT, it requires the ICL ports to be strictly tagged. The port loop detection feature supports strictly tagged ports.

Loop detection for specific VLAN on a port

Strict mode loop detection can be configured on a specific VLAN for a given port. To configure loop detection on VLAN 10 for interface 1/1, enter a command such as the following.

```
device(config-if-e1000-1/1)#loop-detection vlan 10
```

Syntax: [no] loop-detection [**vlan** *vlan_id*]

Where **vlan-id** enables Loose Mode configuration for a VLAN group.

A port can be tagged or untagged member of this VLAN.

Multiple VLANs can have loop detection configured for a given port. Loop detection BPDUs will be sent out of each configured VLAN on that port.

Loop detection shutdown-disable

Use the **loop-detection shutdown-disable** command to disable the port shutdown feature in case of loop detection. This feature will ensure that the ICL stays up when a loop detection PDU is received on the ICL. This command will be applied to both strict mode or loose mode loop detection. To configure **loop-detection shutdown-disable** to shutdown port 1/1 used for the ICL link, enter a command such as the following.

```
device(config-if-e1000-1/1)#loop-detection shutdown-disable
```

Syntax: loop-detection shutdown-disable

Loop-detection shutdown-sending-port

By default, the receive-port is shutdown by loop detection. The **loop-detection shutdown-sending-port** command will shutdown the port that sent the loop detection PDUs instead of shutting down the receiving port. This will ensure that the ICL stays up when a loop detection PDU is received on the ICL.

This feature is only applicable to strict mode loop detection.

```
device(config-if-e1000-1/1)#loop-detection shutdown-sending-port
```

Syntax: [no] loop-detection shutdown-sending-port

Loop-detection-syslog-duration

If any of the ports has shutdown disabled, any loop detection will be logged into the syslog. Since the port is not shutdown, loop detect PDUs will come at a very fast rate and entries into the syslog are throttled.

By default, syslog-duration is 10 minutes. The configurable range is from 10 minutes to 1440 minutes.This is a global command and any changes will be applied to all interfaces.To configure **loop-detection-syslog-duration** for every 30 minutes, enter a command such as the following.

```
device(config)# loop-detection-syslog-duration 30
```

Syntax: [no] loop-detection-syslog-duration *mins*

The *mins* parameter specifies the configurable range which is from 10 minutes to 1440 minutes.

MCT failover scenarios

1. ICL interface or CCP goes down (Keep alive configured)

   When the keepalive VLAN is used and finds the cluster nodes reachability when the ICL or CCP goes down. If the peer node is reachable over keepalive VLAN, the MCT nodes perform performs the Master/Slave negotiation per client. After negotiation, the Slave shuts down its client ports whereas the Master client ports continue to forward the traffic.

   The Master/Slave negotiation is done per MCT client on the basis of RBridge Id and client Local or Remote reachability. If the client is reachable from both MCT nodes, the higher RBridge Id becomes the Master. If client is reachable from one of the MCT nodes, only then the node on which it is reachable becomes the Master.

   If the peer is not reachable over the keepalive VLAN, then both cluster nodeswill keep forwarding.

   NOTE
   Brocade recommends to use keepalive VLANs with the MCT configurations. This will provide a backdoor reachability ifthe ICL interface goes down.

2. ICL interface or CCP goes down (Keep alive not configured)

   When the keepalive VLAN is not configured, both cluster nodes will keep forwarding. Use the **client-isolation strict** command to remove the client interface as soon as ICL goes down and isolate the client completely

3. MCT node goes down.

   When the MCT nodes goes down, the traffic will failover to the other MCT node.

4. Hitless failover performed on one of the MCT nodes

   Traffic is switched over to the other node. However, the CCP will go down and come back up again once the hitless failover is completed.

   Use the **client-interfaces shutdown** command to shutdown all the client interfaces so that the traffic failovers to the other MCT node first. Then perform the hitless failover.

5. Client interface on one of the MCT node goes down

   When hitless failover happens on a Brocade MLXe, NetIron MLX, or NetIron XMR node, that node flushes all the MACs and will reestablish cluster CCP session. In this case, the user may notice some traffic impact.

6. Double failures - The ICL goes down and client interface goes down on one MCT node.

   Multiple failures could drop traffic in this scenario even if there is actual physical path is available.

Show commands

Use the **show cluster** command to display the peer and client states.

```
device#show cluster
Cluster CLUSTER-1 2000
==================
Rbridge Id: 35535, Session Vlan: 2001, Keep-Alive Vlan: 301
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range: 2 to 2000 2002 to 4090
Active Member Vlan Range: 2 to 3 21 to 148 201 404 to 445 501 to 508 2010 3511 to 3574 4021 to 4025 4051
4070 4080 4087 4090
 ICL Info:
 ---------
```

```
 Name            Port  Trunk
 ICL-1           2/1   6
 Peer Info:
 ----------
 Peer IP: 1.1.1.1, Peer Rbridge Id: 1, ICL: ICL-1
 KeepAlive Interval: 50 , Hold Time: 300, Fast Failover
 Active Vlan Range:  2 to 3 21 to 148 201 404 to 445 501 to 508 2010 3511 to 3574 4021 to 4025 4051 4070
4080 4087 4090
 Peer State: CCP Up (Up Time:   0 days:19 hr:24 min: 8 sec)
 Client Info:
 ------------
 Name            Rbridge-id Config     Port  Trunk FSM-State
 Client1         2222       Deployed   1/2     3   Up
 Client2         222        Deployed   1/40    -   Up
```

Syntax: show cluster

Use the **show cluster client** command to display additional State Machine information including the reason for Local CCEP down.

```
device#show cluster mct client c2
...
State: Remote Up
Reason for Local CCEP down: "client-interfaces shutdown
" command
Number of times Local CCEP down: 2
Number of times Remote CCEP down: 1
Number of times Remote Client undeployed: 1
Total CCRR packets sent: 12
Total CCRR packets received: 13
```

Syntax: show cluster client

The following reasons are displayed for Local CCEP down.

TABLE 76 Reason for Local CCEP down

Reason for Local CCEP down	means....
client-interfaces shutdown	command is configured
client-isolation strict	command is configured
Deploy mismatch	Client is not deployed remotely
Slave state	Client is in Slave State when CCP is down
cluster and client undeployed	Neither the Cluster or Client is deployed.
cluster undeployed	Cluster is not deployed
client undeployed	Client is not deployed

Syslogs and debugging

The following system log messages are displayed when the remote Cluster Client Edge Port (CCEP) state is changed or the remote client is deployed or undeployed.

```
SYSLOG: Jun 1 15:43:36:<14>Jun 1 15:43:36 CES, CLUSTER FSM: Cluster mct (Id: 1), client c2 (RBridge Id: 4) -
Remote client deployed
SYSLOG: Jun 1 16:04:24:<14>Jun 1 16:04:24 CES, CLUSTER FSM: Cluster mct (Id: 1), client c2 (RBridge Id: 4) -
Remote client CCEP up
```

The following system log messages are displayed during the LACP delay for different states of the CCEP.

```
Aug 23 23:19:48:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP up
Aug 23 23:19:48:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Local client CCEP up
Aug 23 23:19:48:I:System: Interface ethernet 4/5, state up
Aug 23 23:19:48:W:LACP: ethernet 4/5 state changes from LACP-BLOCKED to FORWARD
Aug 23 23:19:43:W:LACP: ethernet 4/5 state changes from DOWN to LACP-BLOCKED
```

```
Aug 23 23:19:48:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP up
Aug 23 23:19:45:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP up
Aug 23 23:19:12:I:CLUSTER FSM: Cluster MCT (Id: 1), client 26 (RBridge Id: 26) - Remote client CCEP down
```

Sample configuration

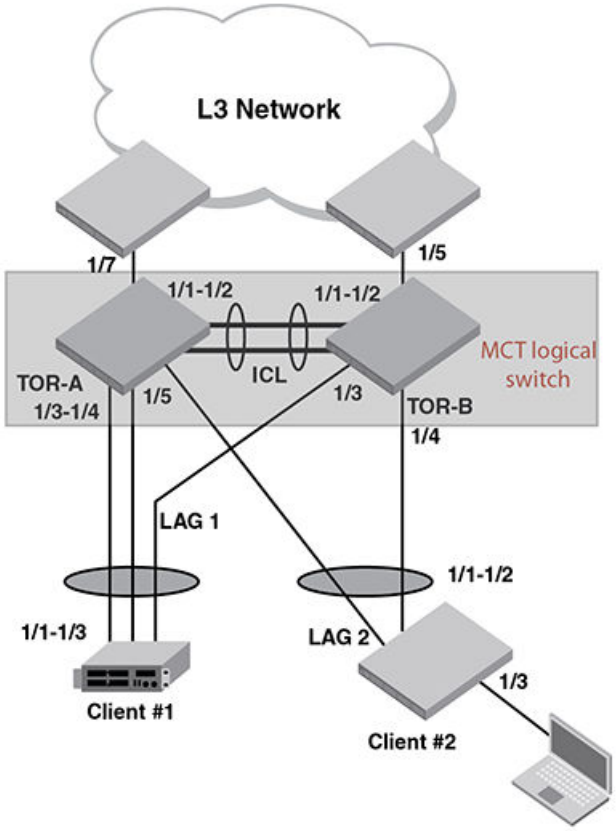
The output below is a sample configuration using port loop detection.

```
device#show run
lag "icl1" dynamic id 1
ports ethernet 3/20 ethernet 4/9
primary-port 3/20
deploy
!
vlan 1 name DEFAULT-VLAN
no untagged ethe 3/20 ethe 4/9
!
vlan 10
tagged ethe 3/20 ethe 4/9
router-interface ve 10
!
vlan 11
untagged ethe 4/17
tagged ethe 3/11 ethe 3/20 ethe 4/9
loop-detection
!
vlan 15
tagged ethe 3/20 ethe 4/9
!
vlan 20
tagged ethe 3/11 ethe 3/20 ethe 4/9 ethe 4/17
!
no route-only
logging console
telnet server
loop-detection-interval 1
loop-detection-disable-duration 1
loop-detection-syslog-duration 11
!
interface ethernet 3/20
loop-detection shutdown-disable
loop-detection vlan 20
!
interface ethernet 4/17
enable
!
loop-detection shutdown-sending-port
loop-detection vlan 20
loop-detection vlan 11
!
interface ve 10
ip address 10.10.10.1/24
!
!
!
cluster abc 1
rbridge-id 100
session-vlan 10
keep-alive-vlan 30
member-vlan 11 to 20
member-vlan 40 to 50
icl icl1 ethernet 3/20
peer 10.10.10.2 rbridge-id 200 icl icl1
client c1
  rbridge-id 300
  client-interface ethernet 3/11
!
```

Failover scenarios for Layer 2 multicast over MCT

Figure 168 shows an example multicast snooping configuration.

FIGURE 165 Multicast snooping over MCT



The following failure modes can occur for Layer 2 multicast over MCT.

Local CCEP down event:

- Outgoing traffic on local CCEP will now go through ICL and go out of remote CCEP.
- Incoming traffic on local CCEP will now ingress through remote CCEP, and then ingress through ICL locally.

Local CCEP up event:

- Outgoing traffic on remote CCEP (after egressing through local ICL) will now start going out of local CCEP.
- Incoming traffic from client through ICL (after ingressing on remote CCEP) will now switch back to local CCEP (this is true only if the client trunk hashing sends the traffic towards local CCEP). CCP (Cluster communication protocol) Down event:
- All related information (i.e. IGMP/MLD group, mcache, router port, static port, pim-sm snooping entry) that was synced from the MCT peer will now be marked for aging locally.

Multicast show commands

Use the **show ip pim mcache** command to display if a CCEP port is being blocked or being forwarded to.

```
device#show ip pim mcache
IP Multicast Mcache Table
Entry Flags    : SM  - Sparse Mode, SSM - Source Specific Mutlicast, DM - Dense Mode
                 RPT    - RPT Bit, SPT - SPT Bit, LSRC - Local Source, LRCV - Local Receiver, JOIN - Join
Upstream
                 HW - HW Forwarding Enabled, FAST - Resource Allocated, TAG - Need For Replication Entry
                 REGPROB - Register In Progress, REGSUPP - Register Suppression Timer
                 MSDPADV - Advertise MSDP, NEEDRTE - Route Required for Src/RP,  PRUN - DM Prune Upstream
Interface Flags: IM - Immediate, IH - Inherited, WA - Won Assert
                 MJ - Membership Join, MI - Membership Include, ME - Membership Exclude
                 BR - Blocked RPT, BA - Blocked Assert, BF - Blocked Filter, BI - Blocked IIF,
            BM - Blocked MCT
Total entries in mcache: 1
1     (2.2.2.101, 239.0.1.3) in v200 (e2/15), Uptime 00:01:08, Rate 42229 (SM)
      Source is directly connected. RP 2.2.2.1
      Flags (0x3046cec1) SM SPT L2REG LSRC LRCV JOIN HW FAST MSDPADV
      fast ports: ethe 2/1
      AgeSltMsk: 00000002,  FID: 0x8006, MVID: NotReq, RegPkt: 0, AvgRate: 41688, profile: none
      Forwarding_oif: 1, Immediate_oif: 1, Blocked_oif: 1
      L2 (HW) 1:
          TR(e2/1,e2/1),  00:01:08/181, Flags: IM IH
      Blocked OIF 1:
          TR(e1/5,e1/5)(VL200), 00:01:08/0, Flags: MJ BM
Number of matching entries: 1
device#
```

The **show ip igmp interface** command has been enhanced to show the IGMP Query suppression state on CCEP ports.

```
device#show ip igmp interface
---------+------+---------+---------------+---------+-----+-----+---------
Intf/Port|Groups| Version |Querier        |  Timer  |V1Rtr|V2Rtr|Tracking
         |      |Oper  Cfg|               |OQrr GenQ|     |     |
---------+------+----+----+---------------+----+----+-----+-----+---------
v200           2    2    -                                      Disabled
    e2/15           2    - Self               0   59 No    No
     e2/1           2    - Self               0   59 No    Yes
     e1/5           2    - Self (MCT-Blk)
     0   40 No    No
device#
```

The IPv6 version of the command **show ipv6 mld interface** has been similarly enhanced.

```
device#show ipv6 mld interface
---------+------+---------+-----------------------------------------+---------+-----+---------
Intf/Port|Groups| Version |Querier                                  |  Timer  |V1Rtr|Tracking
         |      |Oper  Cfg|                                         |OQrr GenQ|     |
---------+------+----+----+-----------------------------------------+----+----+-----+---------
v62            0    2    2                                                      Disabled
     e2/1           2    - Self (MCT-Blk)                          0   79 No
    e1/37           2    - Self                                    0  108 No
    e1/33           2    - Self                                    0  108 No
device#
```

MAC operations

This section describes MAC related configuration operations

MAC Database Update (MDUP)

The MACs that are learned locally are given the highest priority or the cost of 0 so they are always selected as best MAC.

Each MAC is advertised with a cost. Low cost MACs are given preference over high cost MACs.

If a MAC moves from a CCEP port to a CEP port, a MAC move message is sent to the peer and the peer moves the MAC from its CCEP ports to the ICL links.

When peer MACs in a cluster become out of sync for any client, you can recover by performing the following tasks, which are described in the following sections:

- Perform periodic health checks on both peers to determine if the MAC table for any client is out of sync.
- Manually or dynamically synchronize the MAC table.
- Display synchronization details using show commands.

Enabling MAC health check

NOTE
Client health check is disabled by default.

To enable client health check for all configured clients in cluster mode, enter the **client-health-check** command.

```
device(config-cluster-TORA)# client-health-check
```

Syntax: [no] client-health-check

Disabling MAC health check

The **no** version of the client-health-check command disables the client health check for all configured clients in cluster mode.

```
device(config-cluster-TORA)# no client-health-check
```

Syntax: [no] client-health-check

Configuring the health check timer

To configure the periodic timer value of the cluster MAC's synchronization, enter the **client-health-check** command. Time values range from 30 through 120 seconds and 60 seconds is the default.

```
device(config-cluster-TORA)#client-health-check timer 60
```

Syntax: [no]client-health-check [**timer**] [*value inseconds*]

Disabling the health check timer

The **no** version of the **client-health-check timer** command sets the timer to the default of 60 seconds.

```
device(config-cluster-TORA)# no client-health-check timer 60
```

Syntax: [no] client-health-check [**timer**] [*value inseconds*]

Enabling dynamic MAC learning

When the MAC learning mode is set to dynamic and when the remote peer learns that remote client MAC entries are out of sync, a MDUP INFO message is dynamically sent for that particular client. If the dynamic learning mode is disabled, the learning mode is set to the default of manual and the MAC entries must be synchronized manually.

To set the MAC learning mode to dynamic, enter the **client-health-check learning-mode** command

```
device(config-cluster-TORA)#client-health-check learning mode dynamic
```

Syntax: [no] client-health-check learning-mode [dynamic | manual]

Disabling dynamic MAC learning

The no version of the **client-health-check learning-mode** command sets the client health check learning mode to the default of manual and disables the dynamic MAC entries' synchronization.

```
device(config-cluster-TORA)#no client-health-check learning mode dynamic
```

Syntax: [no] client-health-check learning-mode [dynamic | manual]

Manually synchronizing MAC entries and MCT peers

When the MAC database becomes asynchronous between the MCT peers, you can manually synchronize MAC entries specific to various combinations of the cluster, client, and VLANs, as follows:

- All interface MAC entries
- VLAN interface MAC entries
- ALL MAC entries
- VLAN MAC entries
- Client MAC entries
- VLAND and Client MAC entries
- All Client MAC entries
- Single MAC entry

Manually synchronizing all interface MAC entries

The **cluster sync-cluster-intf-mac** command synchronizes all interface MAC entries between the MCT peers of a cluster.

```
device#Cluster sync-cluster-intf-mac
Cluster - All Interface MAC's Requested from peer
device#
```

Manually synchronizing VLAN interface MAC entries

The **cluster sync-cluster-intf-mac vlan-id** command synchronizes the interface MAC entries for the specified VLAN ID between MCT peers. Basic validation occurs to ensure the VLAN ID is within the valid range.

```
device#Cluster sync-cluster-intf-mac vlan-id 3
Cluster - Interface MAC's Requested from peer for vlan-id: 3
device#
```

Manually synchronizing all MAC entries

The **cluster sync-cluster-mac** command synchronizes the entire MAC database between MCT peers.

```
device#Cluster sync-cluster-mac
Cluster -  All Macs Requested from peer
device
```

Manually synchronizing VLAN MAC entries

The cluster sync-cluster-mac vlan-id command synchronizes all MAC entries associated with the specified VLAN between MCT peers.

```
device#Cluster sync-cluster-mac vlan-id 5
Cluster - Macs Requested from peer for Vlan: 5
device#
```

Manually synchronizing client MAC entries

The **cluster sync-cluster-mac client-rbridge-id** command synchronizes all MAC entries associated with the specified client between MCT peers.

```
device#Cluster sync-cluster-mac client-rbridge-id 100
Cluster - Macs Requested from Peer for client rbridge-id: 100
```

Manually synchronizing VLAND and client MAC entries

The **cluster sync-cluster-mac vlan-id** and **client-rbridge-id** commands synchronize all MAC entries associated with the specified client and a VLAN between MCT peers.

```
device#cluster sync-cluster-mac vlan-id 5 client-rbridge-id 200
Cluster Macs Requested from peer for Vlan: 5 and client rbridge-id: 200
```

Manually synchronizing all client MAC entries

The **cluster sync-cluster-mac client-all** command synchronizes all configured client MAC entries between MCT peers.

```
device#Cluster sync-cluster-mac client-all
Cluster - Macs Requested from Peer for all Clients
```

Manually synchronizing a single MAC entries

The **cluster sync-cluster-mac mac** command synchronizes the specified MAC entry specific to a VLAN from the MCT peer.

```
device#Cluster sync-cluster-mac mac 001b.eda4.1d41 vlan-id 8
Cluster - Mac Update Requested from peer
```

Set the client-interfaces delay value

Use the **client-interfaces delay** command to set the delay before brining up the CCEP port. This command is used to set the delay, so that after a node is reloaded, with just L2vpn peer alone, the delay to bring up the CCEP port will be the designated value.

```
device(config-cluster-TOR)#client-interfaces delay 60
```

Syntax: [no] client-interfaces delay *time in sec*

The default value for delay is 30 seconds. The acceptable values range between 20 to 600 seconds.

> NOTE
> Client-interface delay is only applied with just L2 VPN. It does not support L2+L2VPN.

Enabling Cluster MAC synchronization

MAC table learning allows periodic synchronization between MCT peers, based on a configured value (in minutes). If cluster MAC synchronization is enabled but no value is specified, the default synchronization value is 15 minutes.

> NOTE
> Cluster MAC synchronization is disabled by default.

To enable cluster MAC synchronization for all configured clients in cluster mode, enter the **cluster-mac-sync** command.

```
device(config-cluster-TORA)# cluster-mac-sync
```

Syntax: cluster-mac-sync

Disabling Cluster MAC synchronization

The **no** version of the **cluster-mac-sync** command disables the cluster MAC synchronization for all configured clients in cluster mode.

```
device(config-cluster-TORA)# no cluster-mac-sync
```

Syntax: [no]cluster-mac-sync

Configuring the Cluster MAC synchronization timer

To configure the timer value of the cluster MAC's synchronization, enter the **cluster-mac-sync** command. Time values range from 5 through 60 minutes and 15 minutes is the default.

```
device(config-cluster-TORA)#cluster-mac-sync timer 45
```

Syntax: cluster-mac-sync [**timer**] [*value inseconds*]

Disabling the Cluster MAC synchronization timer

The **nocluster-mac-sync** command sets the timer to the default of 15 minutes.

Syntax: [no] cluster-mac-sync [**timer**] [*value inseconds*]

Cluster MAC types

Cluster Local MAC (CL): MACs that are learned on VLANs that belongs to cluster VLAN range and on CEP locally.

MACs are synchronized to the cluster peer and are subject to aging.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/1 0 20 CL Default_ESI
```

Cluster Remote MAC (CR): MACs that are learned via MDUP message from the peer (CL on the peer) The MACs are always programmed on the ICL port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for these MACs with a cost of 1, and associated with the peer rbridge id.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CR
Default_ESI
```

Cluster Client Local MAC (CCL): MACs that are learned on VLANs that belongs to cluster VLAN range and on CCEP ports.

The MACs are synchronized to the cluster peer and are subject to aging. A MDB entry is created for these MACs with a cost of 0 and are associated with the client and cluster rbridge IDs.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCL
Default_ESI
```

Cluster Client Remote MAC (CCR): MACs that are learned via MDUP message from the peer (CCL on the peer) The MACs are always programmed on the corresponding CCEP port and do not age. They are deleted only when it is deleted from the peer. A MDB entry is created for the MACs with the cost of 1, and are associated with the client and peer rbridge ids.

```
device#show mac
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCR
Default_ESI
```

MAC aging

Only the local MAC entries are aged on a node. The remote MAC entries will be aged based on explict MDUP messages only.

The remote MACs learned through MDUP messages are dynamic MACs with the exception that they never age from FDB.

MAC flush

If the CEP port is down, the MACs are flushed and individual MAC deletion messages are sent to the Peer.

If the CCEP local port is down, the MACs are flushed locally and individual MAC deletion messages are sent to peer.

If the **clear mac** command is given, all the MDB and FDB are rebuilt.

If the **clear mac vlan** command is given, all the local MDB and FDB are rebuilt for that VLAN.

MAC movement happens normally on the local node.

CEP to CCEP MAC movement - MAC movement normally happens on the local node, and deletes all the other MDBs from the peer to create a new local MDB.

CCEP to CEP MAC movement - MAC movement happens normally on the local node and delete all the other MDBs from the peer to create a new local MDB.

Flooding support on VLANs

Brocade support the existing VLAN hardware flooding features such as unknown-unicast-flooding and vlan-cpu-protection on cluster VLANs. However, some changes were made to the way CAM entries are programmed. To support MCT cluster VLANs, the following changes to how the CAM is programmed:

- If the ICL port is part of a PPCR, then the device will program the specific (port or VLAN) based hardware flooding CAM entries on that PPCR. This is to avoid duplicate hardware flooding packets to be sent to CCEP ports.
- On an ICL port, the FID in pram will point to MCT_VLAN_CCEP_CONTROL_FID
- On non-ICL port, the FID in pram will point to VLAN_FID

Handling the MAC mismatch scenario in MCT

To handle a MAC address mismatch in the Layer 2 Ethernet header and the ARP sender MAC address in MCT, configure the static MAC in the CCEP port to avoid the traffic impact.

When the CCEP port goes down, the configuration moves the static MAC address from CCEP to the ICL port during a CCEP port shutdown. When the CCEP is up, the static MAC address moves the static MAC address from the ICL port to the CCEP.

The MAC mismatch configuration is supported on the Brocade NetIron XMR Series, Brocade NetIron MLX Series, Brocade NetIron CER Series, and the Brocade NetIron CES Series platforms.

Configuration steps

1. Set up the MCT topology and ensure that the MCT cluster is up and running.

2. Configure the static MAC address on the local CCEP similar to the following example.

```
device(config)# vlan 200
device(config)# static-mac-address 0001.2222.2323 ethernet 1/1
```

3. Enter the **cluster-client-static-mac-move** command for static MAC address movement on both the MCT peer switches in the cluster.

The syslog message helps you identify the root cause for the traffic outage scenario and you can proceed with the static MAC address workaround in MCT by configuring the static MAC address in the CCEP port. The following syslog message is displayed when there is a MAC address mismatch.

```
SYSLOG: <14>Dec 16 05:53:23 MLX_1 MAC_MISMATCH_DETECTION: ARP pkt received with diff eth source MAC
and diff ARP sender MAC. Eth src MAC: 0024.3892.4c02 ARP sender MAC: 0034.2867.2c01.
```

Show Commands

To display all MAC entries, use the **show mac** command as shown below:

```
device# show mac
Total active entries from all ports = 120000
Type Code - ST:Static SEC:Secure 1x:Dot1x NA: NotAvail A:Allow D:Deny
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
Port Type - CEP:Customer Edge PNP:Provider Network BEP:Backbone Edge
BNP:Backbone Network
Vlan Type - C:Customer S:Service B:Backbone I:ISID
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0101.84ff 1/13 0 20 CCL
Default_ESI
0000.0100.4780 1/13 0 20 CCR
Default_ESI
0000.0800.0663 1/14 0 20 CL
Default_ESI
0000.0800.0870 1/1 0 20 CR
Default_ESI
```

Syntax: show mac

To display all the Cluster Local MAC entries for a cluster, use the **show mac cluster** command as shown below:

```
device#show mac cluster abc
Total Cluster Enabled(CL+CR+CCL+CCR) MACs: 451
Total Cluster Local(CL) MACs: 100
Total Cluster Remote(CR) MACs: 151
Total Cluster Client Macs(CCL+CCR) for all clients: 200
Total Cluster Client Local(CCL) MACs for all clients: 200
CCL: Cluster Client Local CCR:Cluster Client Remote CL:Local CR:Remote
MAC Address Port Age VLAN Type FDID TNID ESI
0000.0700.4b04 1/13 0 20 CCL
Default_ESI
0000.0800.3500 1/13 0 20 CCR
Default_ESI
```

Syntax: show mac [**cluster** *id* | *name* **local** | **remote**]

Clear MAC commands

To clear all MACs in the system, enter a command such as the following.

```
device#clear mac
```

Syntax: clear mac

Clear cluster specific MACs

To clear cluster specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster TOR 1 local
```

Syntax: clear mac cluster *cluster-id* | *cluster-name* { **local** | **remote** }

Clear client specific MACs

To clear client specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster TOR 1 client 1 local
```

Syntax: clear mac cluster *cluster-id* | *cluster-name* **client** *client-name* { **local** | **remote** }

Clear VLAN specific MACs

To clear VLAN specific MACs in the system, enter a command such as the following.

```
device#clear mac vlan 2
```

Syntax: clear mac vlan *vlan_id*

Clear cluster VLAN specific MACs

To clear cluster VLAN specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster cluster TOR 1 vlan 1 local
```

Syntax: clear mac cluster *cluster_id* | *cluster-name* **vlan** *vlan_id* { **Local** | **Remote** }

Clear cluster client vlan specific MACs

To clear cluster client specific MACs in the system, enter a command such as the following.

```
device#clear mac cluster TOR 1 vlan 2 client client 1 local
```

Syntax: clear mac cluster *cluster_id* | *cluster-name* **vlan** *vlan_id* **client** *client_name* { **Local** | **Remote** }

Displaying MDUP packet statistics

To display the statistics of MDUP packets, enter a command such as the following.

```
device#show mac mdup-stats
MDUP Information
================
MDUP Data buffers in queue : 0
MDUP Statistics
===============
MDUP Update Messages sent: 7
Add Mac sent: 20
Del Mac sent: 0
Move Mac sent: 0
MDUP Mac Info Messages sent: 1
MDUP Flush Messages sent: 1
```

```
MDUP Synch Messages sent: 0
MDUP Update Messages received: 3
Add Mac received: 40
Del Mac received: 0
Move Mac received: 0
MDUP Mac Info Messages received: 0
MDUP Flush Messages received: 0
MDUP Synch Messages received: 0
```

Syntax: show mac mdup-stats

Clearing the statistics of MDUP packets

To clear the statistics of MDUP packets, enter a command such as the following.

```
device# clear mac mdup-stats
```

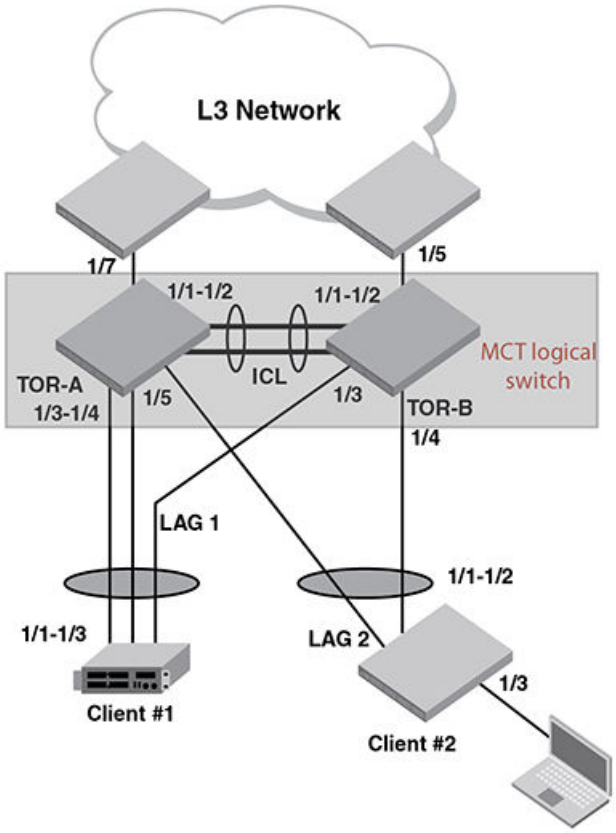
Syntax: clear mac mdup-stats

MCT configuration examples

The following examples displays the module provisioning information from a configuration file:

Single level MCT example

FIGURE 166 Single level MCT



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-B:1/1" ethernet 1/1
 port-name "ICL-to-TOR-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/1" ethernet 1/3
 port-name "lag-client-1:1/2" ethernet 1/4
!
lag "3" dynamic id 3
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-2:1/1" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
```

```
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/7
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 port-name CEP-PC
 enable
!
interface ethernet 1/7
 port-name to-L3-ECMP
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/5
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-A:1/1" ethernet 1/1
 port-name "ICL-to-TOR-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/3" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4
 primary-port 1/4
 deploy
```

```
 port-name "lag-client-2:1/2" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ethernet 1/5
 port-name to-L3-ECMP
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

Client-1:

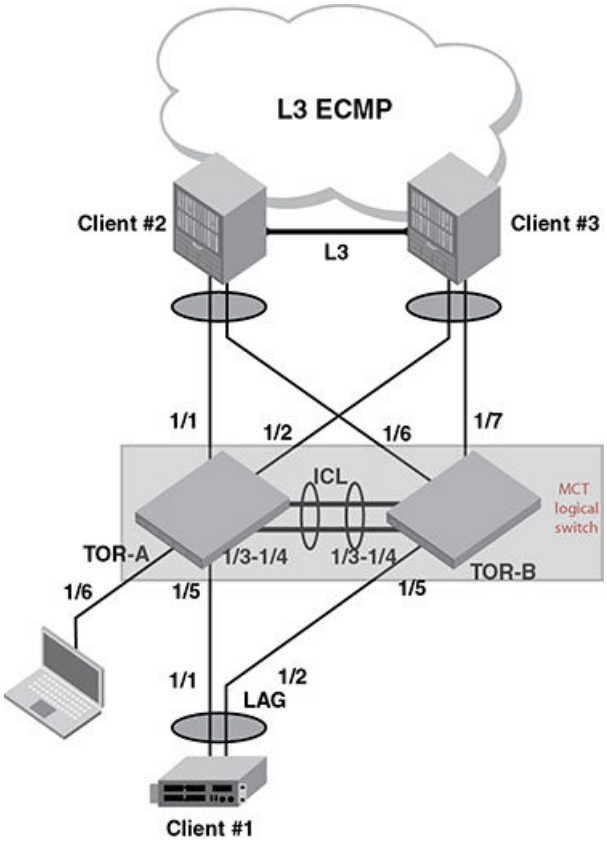
```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/3
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-A" ethernet 1/2
 port-name "lag-to TOR-B" ethernet 1/3
!
interface ethernet 1/1
 enable
!
end
----------------------------------------
```

Client-2:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name to-Host-PC
 enable
!
end
```

Single level MCT- extension example

FIGURE 167 Single level MCT- extension



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1
 primary-port 1/1
 deploy
```

```
 port-name "lag-client-2:1/1" ethernet 1/1
!
lag "2" dynamic id 2
 ports ethernet 1/2
 primary-port 1/2
 deploy
 port-name "lag-client-3:1/1" ethernet 1/2
!
lag "3" dynamic id 3
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "ICL-to-TOR-B:1/3" ethernet 1/3
 port-name "ICL-to-TOR-B:1/4" ethernet 1/4
!
lag "4" dynamic id 4
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-1:1/1" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/3 to 1/4
!
vlan 2 name client-VLAN
 untagged ethe 1/1 to 1/2 ethe 1/5 to 1/6
 tagged ethe 1/3 to 1/4
!
vlan 4090 name Session-VLAN
 tagged ethe 1/3 to 1/4
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/2
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 port-name CEP-PC
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/3
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/5
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
 client Client-3
```

```
  rbridge-id 300
  client-interface ethernet 1/2
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/6
 primary-port 1/6
 deploy
 port-name "lag-client-2:1/2" ethernet 1/6
!
lag "2" dynamic id 2
 ports ethernet 1/7
 primary-port 1/7
 deploy
 port-name "lag-client-3:1/2" ethernet 1/7
!
lag "3" dynamic id 3
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "ICL-to-TOR-A:1/3" ethernet 1/3
 port-name "ICL-to-TOR-A:1/4" ethernet 1/4
!
lag "4" dynamic id 4
 ports ethernet 1/5
 primary-port 1/5
 deploy
 port-name "lag-client-1:1/2" ethernet 1/5
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/3 to 1/4
!
vlan 2 name client-VLAN
 untagged ethe 1/5 to 1/7
 tagged ethe 1/3 to 1/4
!
vlan 4090 name Session-VLAN
 tagged ethe 1/3 to 1/4
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 enable
!
interface ethernet 1/6
 enable
!
interface ethernet 1/7
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/3
 peer 1.1.1.1 rbridge-id 1 icl TOR
```

```
 deploy
 client Client-1
  rbridge-id 100
  client-interface ethernet 1/5
  deploy
 client Client-2
  rbridge-id 200
  client-interface ethernet 1/6
  deploy
 client Client-3
  rbridge-id 300
  client-interface ethernet 1/7
  deploy
!
end
----------------------------------------
```

Client-1:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
interface ethernet 1/1
 enable
!
end
----------------------------------------
```

Client-2:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
vlan 2
 untagged ethe 1/1 to 1/3
 router-interface ve 2
!
router ospf
 area 0
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name L3-ECMP-Cloud
!
interface ve 2
 ip address 10.10.10.1/24
 ip ospf area 0
!
end
----------------------------------------
```

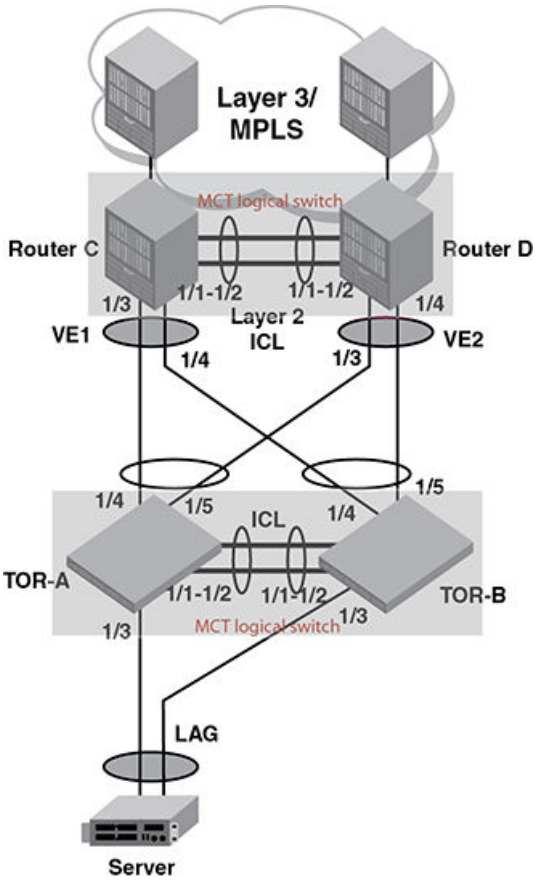
Client-3:

```
!
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
```

```
 deploy
 port-name "lag-to TOR-A" ethernet 1/1
 port-name "lag-to TOR-B" ethernet 1/2
!
vlan 2
 untagged ethe 1/1 to 1/3
 router-interface ve 2
!
router ospf
 area 0
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 port-name L3-ECMP-Cloud
!
interface ve 2
 ip address 10.10.10.2/24
 ip ospf area 0
!
end
```

Two level MCT example

FIGURE 168 Two level MCT



TOR-A:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-B:1/1" ethernet 1/1
 port-name "ICL-to-TOR-B:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-Server:1" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4 to 1/5
 primary-port 1/4
 deploy
 port-name "lag-Router-C:1/3" ethernet 1/4
 port-name "lag-Router-D:1/3" ethernet 1/5
```

```
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-A
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster TOR 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl TOR
 deploy
 client Server-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Routers
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

TOR-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-TOR-A:1/1" ethernet 1/1
 port-name "ICL-to-TOR-A:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3
 primary-port 1/3
 deploy
 port-name "lag-client-Server:2" ethernet 1/3
!
lag "3" dynamic id 3
 ports ethernet 1/4 to 1/5
 primary-port 1/4
 deploy
 port-name "lag-Router-C:1/4" ethernet 1/4
 port-name "lag-Router-D:1/4" ethernet 1/5
!
no route-only
```

```
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/4
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster TOR 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl TOR ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl TOR
 deploy
 client Server-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
 client Routers
  rbridge-id 200
  client-interface ethernet 1/4
  deploy
!
end
----------------------------------------
```

Router C:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Router-D:1/1" ethernet 1/1
 port-name "ICL-to-Router-D:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-TOR-A:1/4" ethernet 1/3
 port-name "lag-TOR-B:1/4" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
```

```
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MPLS-Cloud
 enable
!
interface ve 100
 ip address 1.1.1.3/24
!
!
cluster Router 2
 rbridge-id 3
 session-vlan 4090
 member-vlan 2
 icl Router ethernet 1/1
 peer 1.1.1.4 rbridge-id 4 icl Router
 deploy
 client TOR
  rbridge-id 1
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

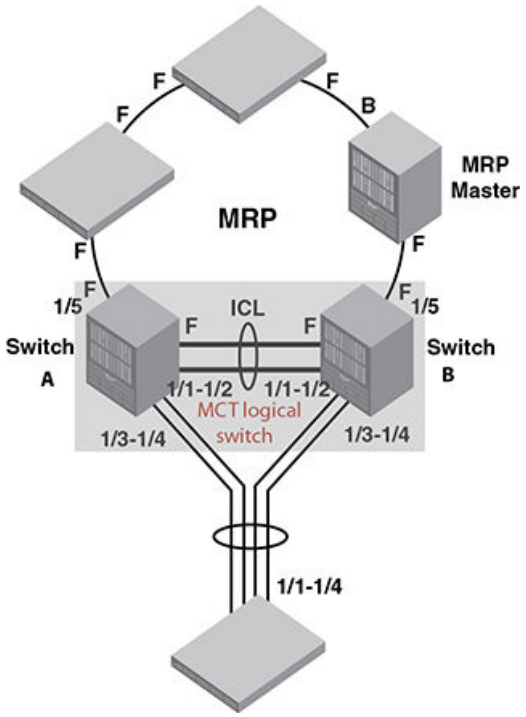
Router D:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Router-C:1/1" ethernet 1/1
 port-name "ICL-to-Router-C:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-TOR-A:1/5" ethernet 1/3
 port-name "lag-TOR-B:1/5" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname TOR-B
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
```

```
!
interface ethernet 1/5
 port-name MPLS-cloud
 enable
!
interface ve 100
 ip address 1.1.1.4/24
!
!
cluster Router 2
 rbridge-id 4
 session-vlan 4090
 member-vlan 2
 icl Router ethernet 1/1
 peer 1.1.1.3 rbridge-id 3 icl Router
 deploy
 client TOR
  rbridge-id 1
  client-interface ethernet 1/3
  deploy
!
end
```

MRP integration with MCT example

FIGURE 169 MRP integration with MCT



MCT-capable-switch-A

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-2:1/1" ethernet 1/1
```

```
 port-name "ICL-to-Switch-2:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/1" ethernet 1/3
 port-name "lag-client-1:1/2" ethernet 1/4
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
 metro-ring-1
  ring-interfaces ethe 1/1 ethe 1/5
  enable
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname Switch-1
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MRP-from-Master
 enable
!
interface ve 100
 ip address 1.1.1.1/24
!
!
cluster MRPRing 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 2
 icl MRPRing ethernet 1/1
 peer 1.1.1.2 rbridge-id 2 icl MRPRing
 deploy
 client client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

MCT-capable-switch-B:

```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/2
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-1:1/1" ethernet 1/1
 port-name "ICL-to-Switch-1:1/2" ethernet 1/2
!
lag "2" dynamic id 2
 ports ethernet 1/3 to 1/4
 primary-port 1/3
 deploy
 port-name "lag-client-1:1/3" ethernet 1/3
 port-name "lag-client-1:1/4" ethernet 1/4
```

```
!
no route-only
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/1 to 1/2
!
vlan 2 name client-VLAN
 untagged ethe 1/3 to 1/5
 tagged ethe 1/1 to 1/2
 metro-ring-1
  ring-interfaces ethe 1/1 ethe 1/5
  enable
!
vlan 4090 name Session-VLAN
 tagged ethe 1/1 to 1/2
 router-interface ve 100
!
hostname Switch-2
!
interface ethernet 1/1
 enable
!
interface ethernet 1/3
 enable
!
interface ethernet 1/5
 port-name MRP-to-Master
 enable
!
interface ve 100
 ip address 1.1.1.2/24
!
!
cluster MRPRing 1
 rbridge-id 2
 session-vlan 4090
 member-vlan 2
 icl MRPRing ethernet 1/1
 peer 1.1.1.1 rbridge-id 1 icl MRPRing
 deploy
 client client-1
  rbridge-id 100
  client-interface ethernet 1/3
  deploy
!
end
----------------------------------------
```

client-Switch:

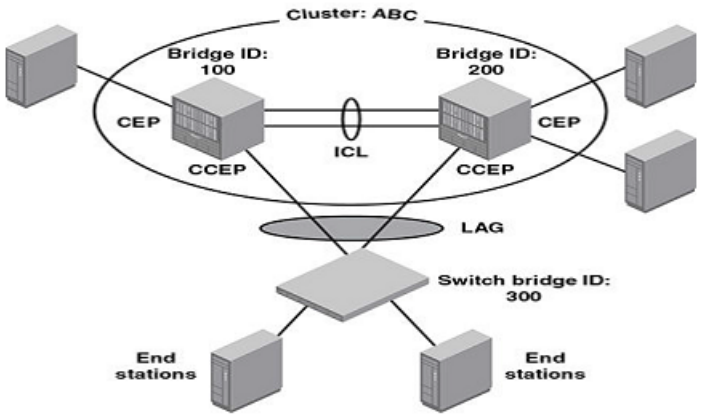
```
lag "1" dynamic id 1
 ports ethernet 1/1 to 1/4
 primary-port 1/1
 deploy
 port-name "ICL-to-Switch-1:1/3" ethernet 1/1
 port-name "ICL-to-Switch-1:1/4" ethernet 1/2
 port-name "ICL-to-Switch-2:1/3" ethernet 1/1
 port-name "ICL-to-Switch-2:1/4" ethernet 1/2
!
interface ethernet 1/1
 enable
!
end
```

Configuring sync CCEP early LACP delay

The **sync_ccep_early lacp-delay** command will ensure that the LACP stays in the LACP-BLOCKED state on the CCEP LAG for the interval configured in the command.

When an CCEP port that was in a Down state on one MCT peer comes Up, the **sync_ccep_early lacp-delay** command will ensure that the LACP stays in the LACP-BLOCKED state on the CCEP LAG for the interval configured in the command. This will give additional time to the other MCT peer to process the REMOTE-UP event. This ensures that the BUM packets received in the interim (between the port being enabled and the REMOTE-UP event being processed) are handled correctly.

FIGURE 170 Configuration steps for a sample network



Configure a cluster on the MCT node as per the network diagram.

Configure the command introduced as part of this feature as follows:

```
device(config-cluster-MCT)# sync_ccep_early lacp-delay 5
```

Sample of the configuration after **sync_ccep_early lacp-delay** command is configured.

```
cluster "ABC" 1
 rbridge-id 100
 session-vlan 99
 icl l2icl ethernet 2/6
 peer 172.16.10.2 rbridge-id 200 icl l2icl
 l2vpn-peer 1.1.1.2 rbridge-id 200
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client "switch bridge"
  rbridge-id 300
  client-interface ethernet 3/13
  deploy
```

Sample **show cluster config** output after the **sync_ccep_early lacp-delay** command has been configured.

```
device(config-cluster-MCT)# show cluster config
cluster "ABC" 1
 rbridge-id 100
 session-vlan 99
 icl l2icl ethernet 2/6
 peer 172.16.10.2 rbridge-id 200 icl l2icl
 l2vpn-peer 1.1.1.2 rbridge-id 200
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client "switch bridge"
  rbridge-id 300
  client-interface ethernet 3/13
  deploy
```

Sample **show cluster** output after the **sync_ccep_early lacp-delay** command has been configured.

```
device(config)#show cluster
Cluster mct 1
=============
Rbridge Id: 100, Session Vlan: 99
Cluster State: Deploy
Early sync of CCEP-UP info to MCT node enabled
lacp delay configured 5
Client Isolation Mode: Loose
Configured Member Vlan Range: 10
Active Member Vlan Range: 10
Total Clients Configured : 1 ( Deployed Clients: 1)

 ICL Info:
 ---------
 Name                                Port  Trunk
 icl                                  2/6   2

 Peer Info:
 ----------
 Peer IP: 1.1.1.2, Peer Rbridge Id: 200, ICL: icl
 KeepAlive Interval: 30 , Hold Time: 90, Fast Failover
 Active Vlan Range:  10
 Peer State: CCP Up (Up Time:   0 days: 0 hr:19 min:12 sec)

 Client Info:
 ------------
 Name               Rbridge-id    Config     Port  Trunk FSM-State
 switch bridge        10           Deployed   3/13    3    Up
```

MCT for VRRP or VRRP-E

One MCT switch is the VRRP or VRRP-E master routerand the other MCT switch is VRRP or VRRP-Ebackup router

The MCT switch that acts as backup router needs to ensure that packets sent to a VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that acts as master router will sync the VRRP-E MAC to the other MCT switch that acts as backup router. Both data traffic and VRRP-E control traffic travel through the ICL unless the short-path forwarding feature is enabled.

L3 traffic forwarding from CEP ports to CCEP ports

Traffic destined to the CCEP ports from the client or CEP ports follow the normal IP routing on both master and backup routers. By default, the best route should not involve the ICL link. Only when the direct link from CEP ports to CCEP ports are down will the traffic be re-routed to pass through ICL link.

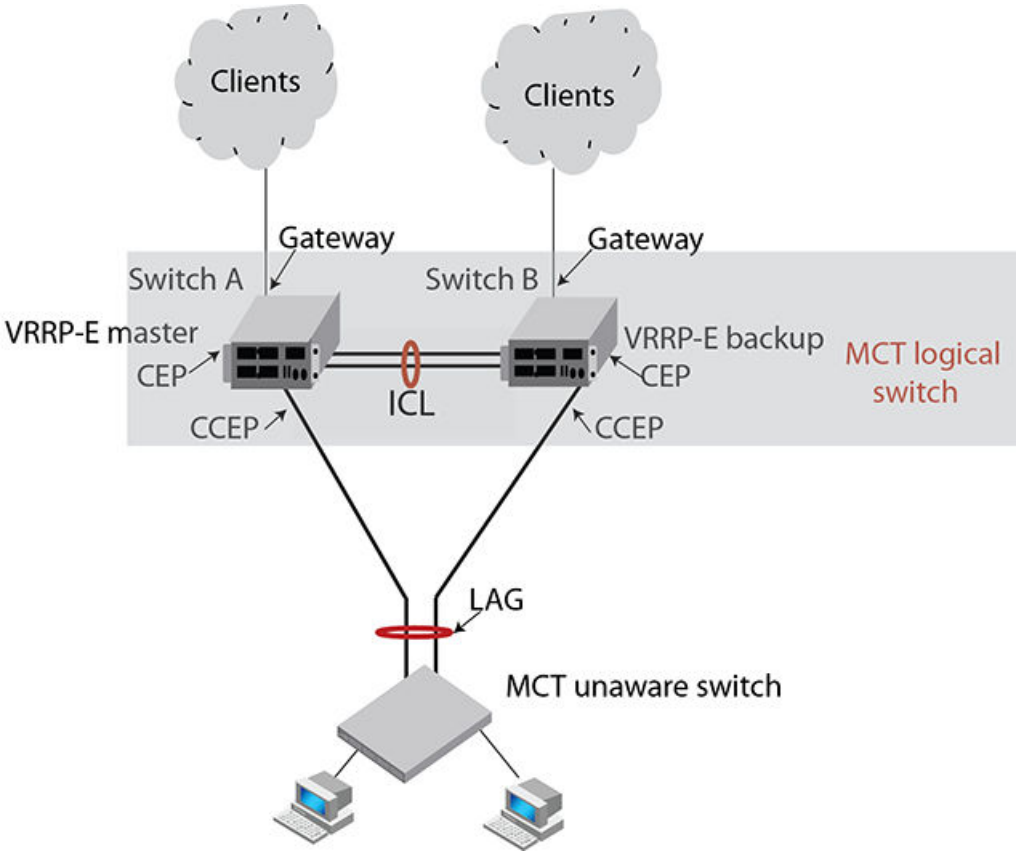
ARP broadcast resolution

Assuming that switch A is VRRP-E master router and switch B is the backup router. ARP request (a broadcast packet) from S1 that is sent through direct link to switch B will be sent to switch A for processing through ICL link. Since MAC learning is disabled on ICL link, the ARP will not be learned automatically through the ICL link. When the ARP request is received by switch A, the reply will be sent through direct link from switch A to S1. If by the time the ARP reply was received the MAC address for the MCT on S1 is not learned yet, the reply packet may be flooded to both the CCEP ports and ICL ports.

Both MCT switches are VRRP or VRRP-E backup routers

In Figure 174, both MCT switches A and B need to ensure packets sent to VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that has direct connection to the master router (who actually learned the VRRP-E MAC from the master) will sync the VRRP-E MAC to the other MCT switch that does not have direct connection to the master. Both data traffic and VRRP-E control traffic travel through ICL unless the short-path forwarding feature is enabled.

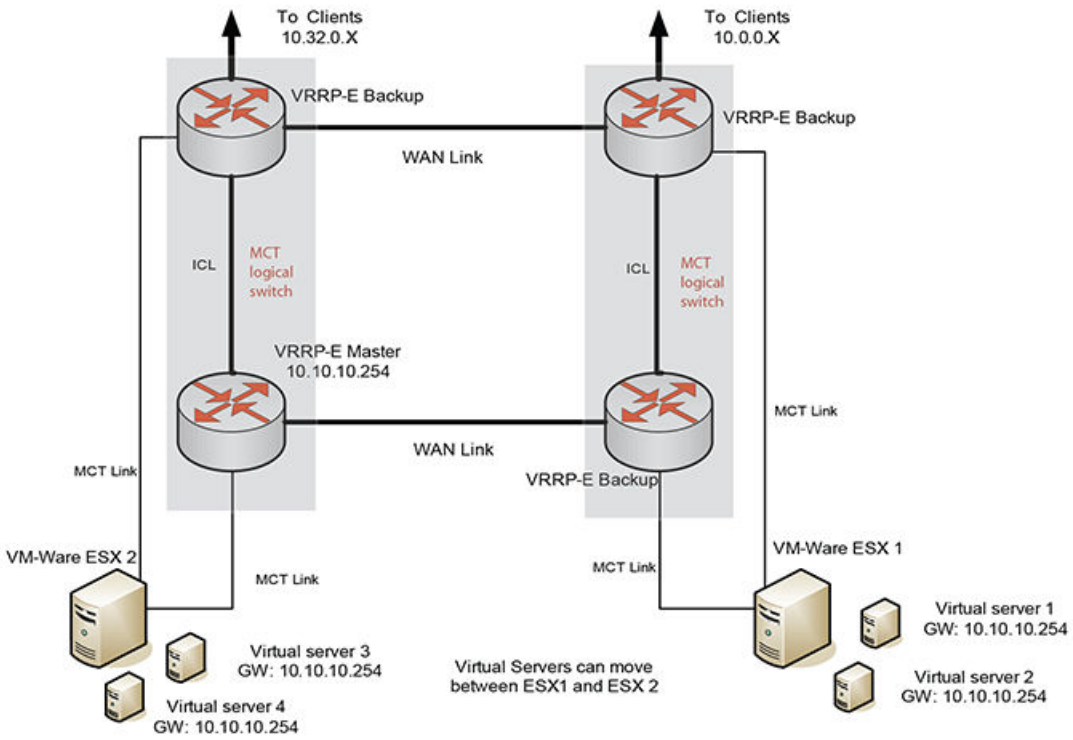
FIGURE 171 Example of MCTS that are Layer 2 switched



In Figure 175, MCTs are deployed on two sites that are connected through two WAN links.

- Two WAN links are completely independent. Switch A and B form MCT 1 and switch C and D form MCT 2. There are L2 protocols running on the VRRP-E routers. L2 protocols will block one of the WAN links to ensure loop-free topology.

FIGURE 172 Example of MCTs that are deployed on two sites that are connected through two WAN links.



Configuration considerations

- VRRE-E virtual MAC will be synced and learned on ICL ports on backup routers through the ICL.
- VRRP or VRRP-E master router will be broadcast hello packets to all VLAN member ports including ICL ports. Normal VLAN FID will be used for broadcasting.
- VRRP or VRRP-E backup routers will not flood back hello packets received from ICL ports to ICL ports, but will be flooded to other non- ICL ports.
- In the current release, MCT switches must have complete routing information using static routes for L3 forwarding.
- For MCT switches configured with VRRP or VRRP-E, track-port features can be enabled to track the link status to the core switches so the VRRP or VRRP-E failover can be triggered.

  NOTE
  Brocade recommends disabling ICMP redirect globally to avoid unintended CPU forwarding of traffic when VRRP or VRRP-E is configured.

L3 traffic forwarding behaviors

When one MCT switch act as VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be L2 switched to the VRRP-E master router for forwarding.
- The VRRP-E MAC will be learned by the other MCT switch that acts as backup router.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

When both MCT devices act as the VRRP or VRRP-E backup routers, the following behavior will be seen:

- Packets sent to VRRP-E virtual IP address will be L2 switched to the VRRP-E master router for forwarding.
- VRRP-E MAC will be learned by both MCT switches acting as backup routers.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

VRRP-E short-path forwarding and revertible option

The **track-port** command will monitor the status of the outgoing port on the backup. It will revert back to standard behavior (no short-path forwarding) temporarily even if short-path forwarding is configured.

Under the VRRP-E VRID configuration level, use the **short-path-forwarding** command. If the revertible option is not enabled, the default behavior will remain the same. Use the following command to enable short path forwarding.

```
device(config-if-e1000-vrid-2)#short-path-forwarding revert-priority
 60
```

Syntax: [no] short-path-forwarding [**revert-priority** *value*]

Use the supplied priority value as a threshold to determine if the **short-path-forwarding** behavior should be effective or not. If one or more ports tracked by the **track-port** command go down, the current priority of VRRP-E will be lowered by a specific amount configured in the **track-port** command for each port that goes down.

Once the current-priority is lower than the threshold, the **short-path-forwarding** will be temporally suspended and revert back to the regular VRRP-E forwarding behavior without **short-path-forwarding** enabled.

The reverting behavior is only temporary. If one or more of the already down ports tracked by the **track-port** command come back, it is possible that the current priority of VRRP-E will be higher than the threshold again and the **short-path-forwarding** behavior will be resumed.

IPv6 VRRP-E short-path forwarding and revertible option

Short-path forwarding enables the short path forwarding on an IPV6 VRRP-E device. It will revert back to standard behavior (no short-path forwarding) temporarily even if short-path forwarding is configured.

Configuration considerations

- VRRP-E virtual MAC will be synced and learned on ICL ports on backup routers through the ICL.
- ICL ports must be member ports of VLANs that CCEP ports are members of.
- VRRP or VRRP-E master router will be broadcast hello packets to all VLAN member ports including ICL ports. Normal VLAN FID will be used for broadcasting.
- VRRP or VRRP-E backup routers will not flood back hello packets received from ICL ports to ICL ports, but will be flooded to other non- ICL ports.
- MCT switches must have complete routing information using static routes for L3 forwarding.
- For MCT switches configured with VRRP or VRRP-E, track-port features can be enabled to track the link status to the core switches so the VRRP or VRRP-E failover can be triggered.

  NOTE
  Brocade recommends disabling ICMP redirect globally to avoid unintended CPU forwarding of traffic when VRRP or VRRP-E is configured.

L3 traffic forwarding behaviors

When one MCT switch act as VRRP or VRRP-E master router and the other MCT switch is VRRP or VRRP-E backup, the following behavior will be seen:

- Packets sent to VRRP-E virtual IPv6 address will be L2 switched to the VRRP-E master router for forwarding.
- The VRRP-E MAC will be learned by the other MCT switch that acts as backup router.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

When both MCT devices act as the VRRP or VRRP-E backup routers, the following behavior will be seen:

- Packets sent to VRRP-E virtual IPv6 address will be L2 switched to the VRRP-E master router for forwarding.
- VRRP-E MAC will be learned by both MCT switches acting as backup routers.
- Both data traffic and VRRP-E control traffic will need to travel through ICL unless the short-path forwarding feature is enabled.

Under the IPv6 VRRP-E VRID configuration level, use the **short-path-forwarding** command. If the revertible option is not enabled, short path forwarding will be disabled if the VRRP-E router priority is below the revert-priority configured value.Use the following command to enable short path forwarding.

Brocade(config-if-e1000-vrid-2)# short-path-forwarding revert-priority 60

Syntax: [**no**] **short-path-forwarding** [**revert-priority** *value*]

Use the supplied priority value as a threshold to determine if the short-path-forwarding behavior should be effective or not. If one or more ports tracked by the track-port command go down, the current priority of IPv6 VRRP-E will be lowered by a specific amount configured in the track-port command for each port that goes down.

Once the current-priority is lower than the threshold, the short-path-forwarding will be temporally suspended and revert back to the regular VRRP-E forwarding behavior without short-path-forwarding enabled.

The reverting behavior is only temporary. If one or more of the already down ports tracked by the track-port command come back, it is possible that the current priority of VRRP-E will be higher than the threshold again and the short-path-forwarding behavior will be resumed.

IPv6 VRRP-E short-path forwarding delay

Use IPv6 VRRP-e short-path forwarding delay to configure the time delay required to enable short path forwarding after reloading the backup router. When configured, short path forwarding will be enabled only after the configured delay time after the MP initialization is completed (from the time all modules in the system are UP). Default value is set to 0 seconds.

This is global IPv6 VRRP-E configuration will effect all IPv6 VRRP-E instances.

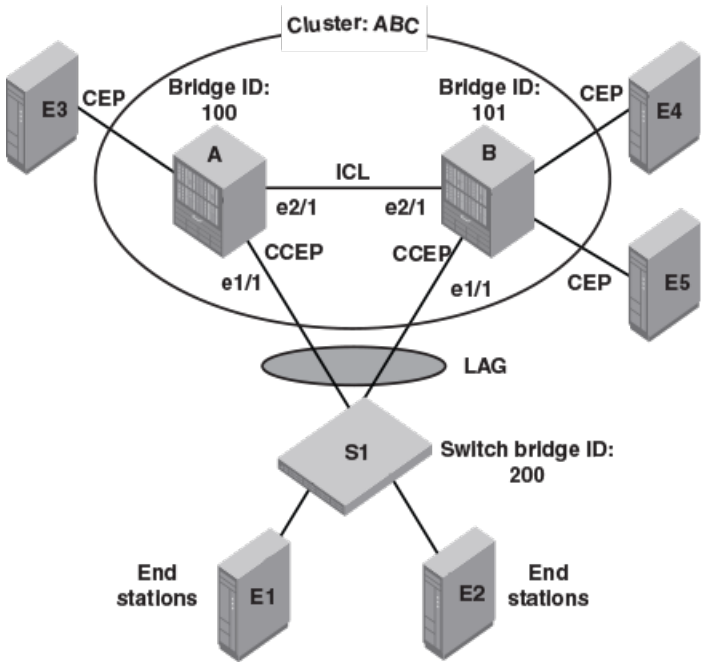
```
device(config)# [no] short-path-forwarding-delay 100
```

Syntax: **short-path-forwarding-delay** *seconds*

Sample configurations

```
device(config)#short-path-forwarding-delay 100
device(config)#ipv6 router vrrp-extended
device(config-ipv6-vrrpe-router)#interface ve 10
device(config-vif-10)# ipv6 address 2003::10:11/64
device(config-vif-10)#ipv6 vrrp-extended vrid 10
device(config-vif-10-ipv6-vrid-10)#backup priority 50
device(config-vif-10-ipv6-vrid-10)#ipv6-address 2003::11:50
device(config-vif-10-ipv6-vrid-10)#short-path-forwarding revert-priority 120
```

Sample MCT Configuration



Switch A:

```
vlan 4090
 tagged ethe 2/1
 router-interface ve 1
!
```

interface ve 1

```
 ip address 192.168.1.1/24
!
```

cluster ABC

```
 rbridge-id 100
 session-vlan 4090
 member-vlan 100 to 300
 icl icl_a_b ethernet 2/1
 peer 10.10.20.2 rbridge-id 101 icl icl_a_b
 deploy
 client switch_s1
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
  exit
!
```

IPv6 VRRP Configuration

```
vlan 200
 tagged ethe 1/1 ethe 2/1
 router-interface ve 10
!
```

Ipv6 router vrrp

```
interface ve 10
 ipv6 address 10::1/64
 ipv6 vrrp vrid 10
  backup priority 50
  ipv6-address 10::100
  activate
!
```

Switch B:

```
vlan 4090
 tagged ethe 2/1
 router-interface ve 1
!
```

interface ve 1

```
ip address 192.168.1.2/24
!
```

cluster ABC

```
rbridge-id 101
 session-vlan 4090
 member-vlan 100 to 300
 icl icl_a_b ethernet 2/1
peer 10.10.20.1 rbridge-id 100 icl icl_a_b
 deploy
 client switch_s1
  rbridge-id 200
  client-interface ethernet 1/1
  deploy
  exit
!
```

IPv6 VRRP Configuration

```
vlan 200
 tagged ethe 1/1 ethe 2/1
 router-interface ve 10
!
```

Ipv6 router vrrp

```
interface ve 10
 ipv6 address 10::2/64
 ipv6 vrrp vrid 10
  backup priority 50
  ipv6-address 10::100
  activate
!
```

NOTE

Cluster client-rbridge-id on both switch A and B have to be same value for a given MCT.

Switch S1:

```
lag "mct_s1" static id 1
 ports ethernet 7/1 to 7/2
 primary-port 7/1
 deploy
!
vlan 200
 tagged ethe 7/1
 router-interface ve 10
!
```

```
interface ve 10
 ipv6 address 10::99/64
```

L2VPN support for L2 MCT clusters

For a L2VPN MCT, L2 MCT peer configuration is not required as it operates independently.

- L2VPN MCT does not require direct ICL as L2VPN may use no-direct MPLS network to the peer MCT node.
- L2VPN MCT does not require L2's session VLAN (and have peer in the same subnet as peer) as peer need not be directly connected.
- L2VPN and L2 MCT can be supported simultaneously
- When L2VPN and L2 needs to be run concurrently, you must configure the L2 peer parameters (similar to MCT/L2) and also configure L2VPN peer as well. The L2 peer will be used for the MCT communication.
- When using the L2VPN service, you must configure a L2VPN peer as well.
- MCT L2VPN is HLOS is not supported but compatible.

Support for non-direct ICL

L2VPN MCT functionality supports non-direct ICL functionality apart from the existing direct ICL requirement that is needed for L2.

A L2VPN MCT session needs to be established or verified at run time so that L2VPN traffic can traverse the ICL path.

The **l2vpn-peer** command needs to be configured to support L2VPN over MCT

A l2vpn peer address is required to be on a remote MCT node loopback address on which the L2VPN sessions is formed.

- This configuration is not mandated by configuration but is required for MCT operation to support L2VPN services.
- A cluster L2 peer address configuration is not required to support L2VPN over MCT.

  NOTE
  For MCT L2VPN services to function, you must configure an operational MPLS tunnel (RSVP or LDP) between the MCT peers.

L2VPN timers

Following optional timers are needed when using MCT L2VPN.

Keep-alive timer : This is used to detect a CCP that is down. The default is 300 milliseconds with hold time of 900 milliseconds. This timer cannot be changed dynamically once the cluster is deployed.

Node-Keep-Alive timer: This is used to detect whether the peer MCT node is down. The default is 2 seconds with hold time of 6 seconds. This timer cannot be changed dynamically once the cluster is deployed. This timer is to quickly detect CCP communication between the MCT peers and to failover quickly. This failure can happen due to route flaps or congestion in the network or in the MCT peer nodes. (Compared to L2 or MCT peer, this is similar to ICL link down).

NOTE
In Brocade NetIron CES and CER devices, the hold time should be configured to a minimum value of 1800 milliseconds to avoid CCP flaps when the **no client-interface shutdown** command is run. For example: `l2vpn-peer 10.6.240.3 timers keep-alive 600 hold-time 1800`

Cluster CCP session rules

NOTE
CCP (Cluster Communication Protocol) can run either on L2VPN peer (in cases where only l2vpn peer configured) or on L2 configured peer (in cases where L2 and L2VPN services need to coexist).

NOTE
If using a L2VPN peer in conjunction with L2 Peer, the L2VPN keepalive timers are not used and we will rely on MCT L2 peer for MCT communication. Additionally, for L2VPN peer case, these timers need to match on both nodes. If not, the CCP using L2VPN will not come up.

L2VPN peer only configurations

The CCP comes up only if:

- CCP session to L2VPN peer has to come up using regular TCP session.

The CCP goes down if either of the below conditions is TRUE:

- L2VPN CCP is down (TCP shuts down the CCP session for some reason).2VPN-CCP-timer expires.

The remote node down event is generated if both scenarios below are TRUE:

- L2VPN-CCP keepalive timer expires. This will indicate route failure (300 milliseconds * 3 ~ 900 milliseconds).
- L2VPN-node keepalive timer expires. This timer is to detect the reachability of the remote node. (Compared to MCT or L2 peer, this is similar to remote MCT node down or not reachable).

  NOTE
  It does not need to be route failure for the timer to expire.

- This timer is to quickly detect CCP communication between the MCT peers and to failover quickly. This failure can happen due to route flaps or congestion in the network or in the MCT peer nodes. (Compared to L2 or MCT peer, this is similar to ICL link down).

L2VPN peer with a L2 peer

The CCP comes up if the following is TRUE:

- Peer normal CCP session comes up

The CCP goes down if the following is TRUE:

- L2 peer CCP session goes down

The remote node down event is generated if the following condition is TRUE:

- L2 CCP session is down and keepalive VLAN does not respond

  NOTE
  CCP is dependent only on L2 in such a configuration

  NOTE
  The behavior for L2 peer only case will be similar to the original L2/MCT implementation.

Handling L2VPN spoke down

For L2VPN/MCT services to function, you must establish a spoke PW between two MCT peers for each L2VPN service instance. This spoke PW will be used for sending traffic from standby MCT node to the remote PE devices and also in failover scenarios. This spoke

PW is like any other PW session between 2 peers and will be established using LDP and can use any MPLS tunnel transport mechanism (RSVP or LDP tunnel).

If the CCP is UP and L2VPN MCT Spoke PW is down, either for all L2VPN Instances or some of the instances then,

- For the L2VPN Instances which are down, a list MCT CCEP clients are collected and trigger the MCT infrastructure to do Master/Slave selection for those MCT clients. This ensures that for each MCT Client only the links connected to one of the MCT node will be UP (Master) and the other MCT node will be down (Slave).
  - This ensures that if MCT Client Ports are NOT shared by L2 and L2VPN then MCT Spoke PW going down will not affect L2.
  - If there are any MCT client ports which are shared by L2 and L2VPN then the MCP Spoke PW going down will affect both L2 and L2VPN.
- When MCT Spoke PW is down and comes up later, it will revert back to active/active MCT clients.

CCP down handling when both L2 and L2VPN exist

When no keep-alive VLAN is configured,

- Client links status will be controlled based on loose and strict configurations. Both links can stay up *or* both links can go down *or* one of them can stay up and one of them can stay down.
- For L2VPN, for both VLL and VPLS, both MCT nodes will take the Active role for signaling towards the remote PEs.

When L2 keep-alive VLAN is configured, the initial state will be same as if there is no keep-alive. Then once keep-alive probes are exchanged, only one of the client links will stay up.

Graceful restart support

Graceful switchover is handled for MCT/L2VPN by the following:

- Using the **client-interfaces shutdown** (cluster configuration) command.
- R1 sends graceful-upgrade-restart (MCT) message to R2 (where the MCT peer nodes is R1 and R2 and R1 needs to be upgraded) and R1 disables all client interfaces locally.
- L2VPN task brings down all the PWs to standby state
- R2 (MCT peer node) performs the following actions:
  - Process the graceful-upgrade-restart (MCT) message that is received via CCP. It will force local L2VPN instances to be forcefully active (though it does not match local configuration).
  - Send back graceful-upgrade-done (MCT) message to R1 to indicate the completion.
  - R1 receives the graceful-upgrade-done message from R2 and it generates a local syslog to indicate the user to continue with the reload/restart operation. User now can proceed with the upgrade.
  - This can minimize the traffic loss in cases where user wants to perform graceful restart operation on one of the MCT nodes.

Show commands

Use the **show cluster** command will display MCT cluster information.

```
device#show cluster
Cluster clu 1
=============
Rbridge Id: 4, Session Vlan: 0
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range:
```

```
Active Member Vlan Range:
show cluster clu client c1
Cluster clu 1
=============
Rbridge Id: 4, Session Vlan: 0
Cluster State: Deploy
Client Isolation Mode: Loose
Configured Member Vlan Range:
Active Member Vlan Range:
Client Info:
------------
L2VPN Peer Info:
---------------
Peer IP: 5.5.5.5, Peer Rbridge Id: 5
KeepAlive Interval: 300 , Hold Time: 900
Node KeepAlive Interval: 2000 , Hold Time: 6000
l2vpn-revertible-timer 300
Peer State: CCP Up (Up Time: 0 days: 0 hr:15 min:18 sec)
Client Info:
------------
Name Rbridge-id Config Port Trunk FSMState
c1 101 Deployed 1/4 2 Admin Up
c2 102 Deployed 1/2 1 Up
```

Syntax: show cluster

See Show commands on page 591 for additional information regarding the **show cluster** command output.

Sample Configurations

To support L2VPN services, the cluster with the newly added L2VPN parameters for this feature must be configured.

Below, the configuration highlighted is the new L2VPN configuration.

Assume: Local MCT node loopback 1.1.1.1; Peer MCT 2.2.2.2; Remote L2VPN Peers: 3.3.3.3, 4.4.4.4;

For L2 operation, the remote peer in direct ICL is assumed to be 10.10.10.2.

Cluster Configuration:

```
cluster mct_l2vpn 1
     rbridge-id 1
     [session-vlan 101]         // The below configuration is needed only for MCT/L2
     [icl interface eth 1/1 icl]
     [peer-address 10.10.10.2 rbridge-id <id>]
     l2vpn-peer 2.2.2.2 rbridge-id <id>
    [l2vpn-peer 2.2.2.2 timers keep-alive <msec> hold-time <msec>]
    [l2vpn-peer 2.2.2.2 timers node-keep-alive <sec> hold-time <sec>]
    deploy

    client MCT_CLIENT1
      rbridge-id 101

 client-interface e 1/1
     [vll-pw-redundancy-active]
      deploy
```

Sample MPLS VLL/VPLS Configuration:

```
router mpls
  vpls-policy
    [vpls-pw-redundancy-active]

  lsp MCT_PEER_LSP
    to 2.2.2.2
    enable
// Below L2VPN configuration is given just for completion
vll MCT_VLL1 101
```

```
 vll-peer 3.3.3.3 [4.4.4.4]
 [vll-pw-redundancy-active]

 vlan 101
   tagged eth 3/1
vpls MCT_VPLS1 101
 vpls-peer 3.3.3.3  4.4.4.4 5.5.5.5  6.6.6.6
 [vpls-pw-redundancy-active]

 vlan 101
          tagged eth 3/1
```

MCT for VPLS

MCT helps organizations build scalable and resilient network infrastructures. MCT is an enhancement over the link aggregation standard, which allows multiple switches to appear as single logical switch connecting to another switch using a standard LAG. MCT is designed to achieve the desired active-active topology and efficient Layer 2 multipathing, while ensuring that the network scales effectively.

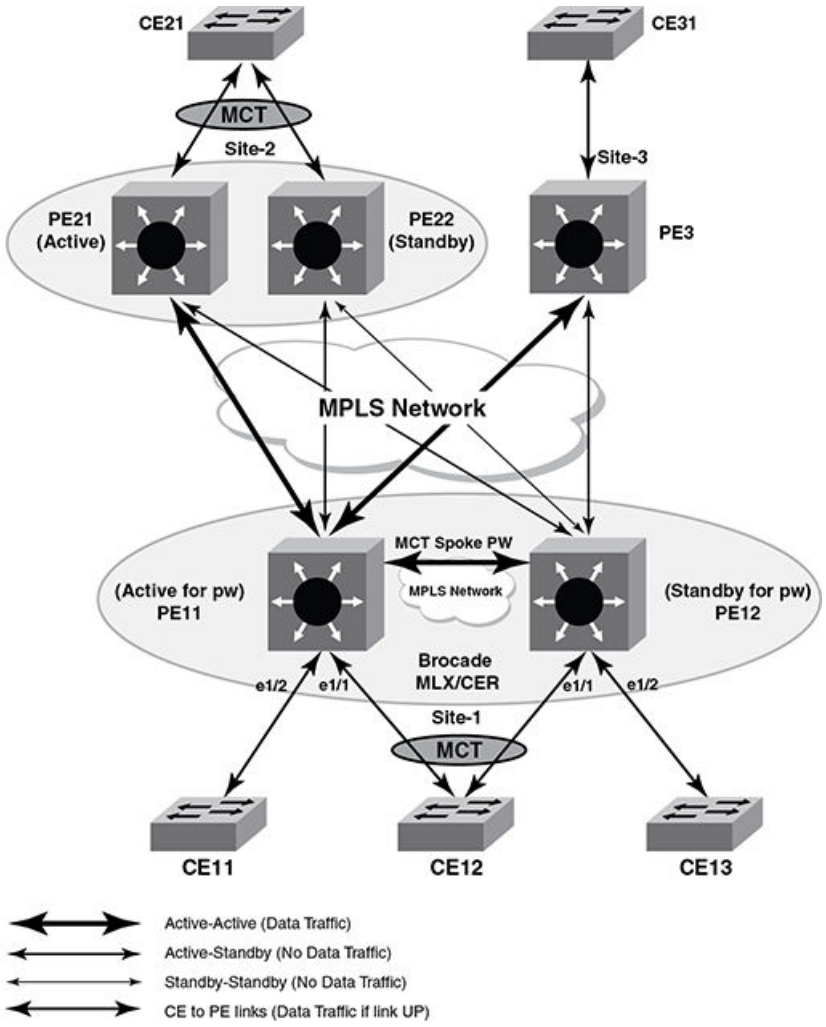
You can connect a customer edge device to an MCT, such as two Brocade MLX's, and then run it over a VPLS network. This is ideal for inter-data center connectivity, or within a campus environment for extending across a backbone. One of the benefits of this is to simplify VM mobility and eliminate single point of failures. For any customers looking to inter-connect multiple data centers and offer new cloud based services, such as disaster recovery, MCT can help you achieve that. In a metro network, MCT helps providers offer their customers enhanced end to end resiliency for business services.

This feature supports dual-homing connectivity of CE devices to PE devices. Dual-homing enables link level and node level redundancy for CE's connected to PE's.

Figure 176 is a typical topology with MCT end-points for VPLS.

- PE11 and PE12 are the two nodes of the MCT Cluster.
- CE12 is connected to the two MCT nodes of the cluster using a LAG. From MCT nodes point of view, the links connected to CE12 are called CCEP end-points.
- CE11 and CE13 are single homed to PE11 and PE12 respectively. These are called the CEP end-points.

FIGURE 173 Sample topology with MCT end-points for VPLS



Configuration Considerations

Certain configurations must be correctly configured on both MCT nodes for MCT to work.

If the following are not configured correctly, then MCT functionality will not come up for that VPLS instance and both MCT nodes will independently come up as Active nodes.

- VPLS instance configuration should be with the same vc-id on both MCT nodes.
- Both MCT nodes should have "MCT" configured for the VPLS Instance.
- Both MCT nodes should have the same vc-mode (tagged or raw mode). If this configuration doesn't match, then the MCT Spoke PW between the two MCT nodes will not come up and MCT functionality will not work.
- Both **vpls-mtu** and **vpls-mtu-enforcement** configuration should be same on both MCT nodes. If this configuration does not match, then the MCT Spoke PW between the two MCT nodes may not come up and MCT functionality will not work.

The following configurations must be same on both MCT nodes for VPLS instance with MCT end-points.

- Same set of remote peers
- Local-switching enable or disable

- VPLS MAC table size

For each VPLS instance which has an MCT end-point, the MCT peer node is configured as a special VPLS peer. This is done on both MCT nodes. This is referred to as the cluster-peer and the PW between them is called the MCT Spoke PW.

One of the nodes from MCT pair is picked as active. This can be done either globally or controlled per VPLS instance.

Active MCT node will establish the PW session with the remote PEs and signal this active status in the status TLV.

MP switchover and Hitless OS upgrade is not supported but compatible.

NetIron CES and NetIron CER limitations

Consider the following limitations for the NetIron CES and NetIron CER devices

- LSP of VPLS VC cannot run over ICL port.
- Spoke VC & other normal VC cannot share same port when local switching is disabled
- As the MCT vpls uses twice the number of MACs in hardware, the **system max mac** shall be reduced to half when all the vpls instances in the node are MCT enabled.

Scalability

The following table provides information about the scalability numbers for the MCT VPLS feature.

TABLE 77 MCT VPLS scalability

Capability	Brocade NetIron MLX Series	Brocade NetIron XMR Series
Maximum number of MCT VPLS instances in the core	4096	16,384
Maximum number of MCT L2VPN instances for MAC addresses	131,072	131,072
VLANs per VPLS or MCT	4096	4096
BUM rate limiting (VPLS CPU protection)	Enabled	Enabled
VPLS VLAN ports and VPLS peers	49,152	49,152

The following limitations impact the MCT VPLS scalability:

- VPLS CPU protection is limited due to the number of Mapped VLAN ID (MVID) resources. Currently, 2000 MVIDs are supported in the system and shared between multiple applications.
- VPLS supports Forwarding ID sharing (except on the BR-MLX-40Gx4-M 4-port, BR-MLX-4-port-10g-M-IPSEC-4-port, BR-MLX-10Gx20 20-port 1/10GbE, and BR-MLX-100Gx2-CFP2 modules).
- Dynamic Forwarding IDs are limited to 8000 in the system and shared between multiple applications.

Forwarding known unicast traffic

On Active MCT PE node, traffic will be forwarded and received from all the remote PE's.

For a host which is single homed to standby MCT PE node, the Active MCT PE node will send the packet on the MCT Spoke PW to standby MCT PE node.

On standby MCT PE node, traffic received from local CCEP and CEP ports will be forwarded to Active MCT PE node through the MCT Spoke PW which will then forward to the remote PE's.

On standby, the MCT PE node traffic received from all other PE nodes except the MCT Spoke PW is dropped.

Forwarding broadcast, unknown unicast, multicast traffic

Known unicast packets will be forwarded based on where the destination MAC is learned (Exception to Packets received on MCT Spoke PW and destined to CCEP) but for BUM traffic it will forwarded to all the destinations listed below.

On the active MCT PE Node

- Traffic received from CE's is forwarded to all Remote PE's, MCT Spoke PW and locally connected CE's.
- Traffic received from all Remote PE nodes (which doesn't include the MCT Spoke PW) is sent to all CE's and a copy is sent to MCT Spoke PW.
- Traffic received from the MCT Spoke PW, is sent to all the Remote PE's and locally connected CE's which are single-homed.

On Standby MCT PE Node

- Traffic received from CE's is forwarded to all locally connected CE's and to MCT Spoke PW which is the only PE session which is active.
- Traffic received from the MCT Spoke PW is forwarded to all locally connected CE's which are single homed to this standby MCT PE node.

MAC Learning and Synching

CCEP endpoints can send traffic to either of the MCT switches. This causes MAC addresses to move from CCEP port and the MCT Spoke PW continuously. To avoid MAC movement between the CCEP port and MCT spoke, MAC learning is disabled on MCT spoke PW and MAC addresses are synced between the two MCT peer switches using MAC database update protocol (MDUP).

MAC addresses learned are added to the MDUP database (MDB). MAC addresses learned locally on the cluster node are added to the local MDB with a cost of zero. MAC addresses learned from the peer cluster node are added to the remote MDB with a cost of one.

If a MAC address exists in both local and remote MDB, the MAC with the lowest cost is selected and added to the forwarding database (FDB). Cluster local MACs are always given preference over cluster remote MACs.

- On Active MCT PE Node MAC's are learned from the packets received from both CE and PE's (except the MCT Spoke PW).
- On standby MCT PE node MAC's are learned from packets received from CE's. (No packets are received from the remote PE's).
- There is no learning of MAC's for packets received on MCT Spoke PW. This is true on both Active and standby PE Node.
- For all VPLS instance which have MCT based endpoints, the complete MAC tables are synched between Master and Slave MCT peers.

MAC Aging

Cluster remote MDB entries are not aged locally. They are deleted only when an MDUP messages is received from the cluster peer switch.

MAC entries are deleted from FDB only if the MAC entry is not present in both local and remote MDBs.

Active-standby role change (revertible timer)

When a MCT node cannot establish a CCP connection with other MCT node, it will declare itself as Active and start the Remote Peering.

When the CCP session is established the node which is already Active will have higher preference over other MCT node if it is in Transit state.

MCT node which is configured to be Active will start the **l2vpn reversion-timer** to take over the Active role (if it is in standby role).

Local switching with MCT

With MCT-VPLS support, local-switching refers to traffic switched between end-points of both the MCT nodes of the cluster.

Identical **vpls-local-switching** configurations are required on the two MCT nodes for either supporting local-switching or disabling local-switching.

When local-switching is disabled, packets from the Active MCT node end-points are not sent to MCT-Spoke PW.

Packets from standby MCT node are always sent to MCT-Spoke PW.

CPU protection with MCT

CPU protection is supported with MCT VPLS.

CPU protection can be turned on independently on each MCT node of the cluster. When the MCT VPLS is enabled for a VPLS instance, it will use **cpu-protection** support with MCT.

Auto-discovery with MCT

VPLS auto-discovery is where the VPLS peer's are discovered by BGP and added into VPLS instead of manually configuring the peers.

If the MCT peer node is discovered as a VPLS peer by BGP, then it will not be added as VPLS remote peer

> NOTE
> This special handling is only for "MCT enabled VPLS Instances" and will not affect non-MCT VPLS instances.

Cluster-peer verses vpls-peer

Cluster-peer PW is the peering session between the two MCT nodes of the cluster. It is called MCT-Spoke PW.

- MCT-Spoke PW is different from the regular PW.
- MCT-Spoke PW bring up will be triggered only when the configuration sync between the two MCT nodes succeeds.
- MCT-Spoke PW signaling is triggered even though there are no local end points that are up or configured.
- MAC learning is disabled in software and hardware for the data traffic received on MCT-Spoke PW.

Graceful Restart and Upgrade

You can gracefully restart a node. If the node which is being brought down for maintenance has an "Active" role for any VPLS instance, that node will become the standby. This can be done using the **client-interface shutdown** command.

NOTE

- When the cluster is configured with **client-interface shutdown** command with only peer [L2] config, all clients on this MCT node are disabled except CCP. However, all data traffic will continue to be forwarded via the other MCT node. This is the known behavior before or after an upgrade.
- When the cluster is configured with **client-interface shutdown** command with l2vpn-peer config, the MCT-SPOKE-PW instance is disabled along with all clients on the specific MCT node. This is the known behavior before or after an upgrade. In this case, the other MCT node moves to MCT VPLS Active state, and all data traffic is forwarded from clients via this MCT node to the remote L2VPN peer.
- When the **client-interface shutdown** command is run to shut down the client interfaces, the Brocade NetIron MLX devices bring down the MCT Cluster Communication Protocol (CCP). However, the Brocade NetIron CER and CES devices do not bring down the CCP during the client interface shutdown.

PE to PE Forwarding

With the support of MCT end-point for VPLS, packets received from a remote PE are sent to cluster-peer (received from a PW and sent to another PW).

Similarly packets received from standby MCT node (using the cluster-cluster peer PW) are sent to Remote PE.

The received MPLS packet are de-capsulated and the L2 Payload is again encapsulated into MPLS packet with the right labels.

Unsupported features for MCT enabled VPLS instances

Following features are not supported for MCT enabled VPLS instances:

- 802.1ag
- IGMP-Snooping.
- VPLS-PBB

Configuring the MCT end-point for a VPLS instance

To enable MCT end-points for a VPLS instance, you must configure the **cluster-peer** for the VPLS Instance. This address should match the **l2vpn-peer** address that is done as part of the cluster configuration.

To configure a VPLS instance with a **cluster-peer** , there should not be any end-points or remote peers configured (auto-discovery should be disabled).

When the **cluster-peer** is configured, VPLS will enable the active or standby status TLV exchange with the remote peer's irrespective of other MCT configuration (like l2vpn-peer, cluster deployed or not deployed, CCEP end-points or no CCEP end-points for this VPLS instance).

If a l2vpn-peer configuration is already done and the cluster-peer configuration doesn't match with l2vpn-peer IP address, then configuration will be rejected.

To enable MCT functionality and to allow adding MCT and CCEP Ports as VPLS End-Points, enter a command such as the following.

```
device(config-mpls)#vpls test 10
deviceXMR4(config-mpls-vpls-test)# cluster-peer 12.12.12.12
```

Syntax: [no] cluster-peer *cluster-peer IPaddress*

The *cluster-peer IP address* parameter specifies the IP address of cluster peer.

The **no cluster-peer** command removes the cluster peer.

NOTE
Before removing the **cluster-peer** configuration for a VPLS instance using the command **no cluster-peer** , all the end points and remote peer configurations must be deleted.

Disabling cluster-peer mode for a VPLS instance error messages

If any end-point is configured while resetting the **cluster-peer** mode for the VPLS instance, the following error message will be displayed.

```
Error: End-point should not be configured while removing Cluster-Peer configuration.
```

If any remote peer is configured while resetting the cluster-peer mode for the VPLS instance, the following error message will be displayed.

```
Error: Remote-peer should not be configured while removing Cluster-Peer configuration.
```

If auto-discover is configured while resetting the cluster-peer mode for the VPLS instance, the following error message will be displayed.

```
Error: auto-discovery should not be configured while removing Cluster-Peer configuration.
```

VPLS global pw-redundancy (optional)

Once MCT is enabled for a VPLS instance, the two MCT cluster nodes synchronize the configuration with each other over the CCP and decide which node will take up the Active role and Standby role. PW redundancy provides is backup PWs ready so that traffic can be quickly failed over to the backup PWs. This command can be configured either globally for all VPLS Instances with MCT or for each VPLS instance individually.

NOTE
If it is not configured, the MCT node with lower rbridge-id will be elected as Active to signal to the remote PE's.

Use the **vpls-pw-redundancy-active** command at the global mode to set the pw-redundancy option for all VPLS Instances with MCT.

```
device(config-mpls)#vpls-policy
device(config-mpls-vpls-policy)# vpls-pw-redundancy-active
```

Syntax: [no] vpls-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

Per VPLS instance pw-redundancy (optional)

If **vpls-pw-redundancy-active** is not configured per VPLS instance, the selection will be based on the global configuration. The per VPLS Instance configuration always has the higher priority over global configuration.

```
device(config-mpls)#vpls test 10
device(config-mpls-vpls-test)# vpls-pw-redundancy-active
```

Syntax: [no] vpls-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

Sample MCT configuration with VPLS endpoints

The following sample configuration shows the two MCT cluster nodes for the topology shown in MCT for VPLS on page 632.

```
Switch PE11:
MLX-PE11# show run
router mpls
 vpls test 10
```

```
  cluster-peer 12.12.12.12
  vpls-peer 21.21.21.21 22.22.22.22 3.3.3.3
  vlan 10
   tag eth 1/1 eth 1/2
……..
cluster abc 1
  rbridge-id 100
  l2vpn-peer 12.12.12.12 rbridge-id 101
  deploy
 client c1
  rbridge-id 300
  client-interface ethernet 1/1
  deploy

Switch PE12:
MLX-PE12# show run
router mpls
 vpls test 10
  cluster-peer 11.11.11.11
  vpls-peer 21.21.21.21 22.22.22.22 3.3.3.3
  vlan 10
   tag eth 1/1 eth 1/2
……..
cluster abc 1
  rbridge-id 101
  l2vpn-peer 11.11.11.11 rbridge-id 100
  deploy
 client c1
  rbridge-id 300
  client-interface ethernet 1/1
  deploy
```

VPLS show commands

```
device# show mpls vpls detail
VPLS test, Id 10, Max mac entries: 2048
Total vlans: 0, Tagged ports: 0 (0 Up), Untagged ports 0 (0 Up)
IFL-ID: n/a
VC-Mode: Raw
Total VPLS peers: 2 (2 Operational)
Cluster-Peer address: 12.12.12.12, State: Operational, Uptime: 2 hr 55 min
Tnnl in use: tnl2(3)[RSVP] Peer Index:0
Local VC lbl: 983040, Remote VC lbl: 983040
Local VC MTU: 1500, Remote VC MTU: 1500
Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
Peer address: 15.15.15.15, State: Operational, Uptime: 2 hr 55 min
Tnnl in use: tnl1(1024)[RSVP] Peer Index:1
Local VC lbl: 983041, Remote VC lbl: 983043
Local VC MTU: 1500, Remote VC MTU: 1500
Local PW preferential Status:Active, Remote PW preferential Status:Active
Local VC-Type: Ethernet(0x05), Remote VC-Type: Ethernet(0x05)
CPU-Protection: ON, MVID: 0x001, VPLS FIDs: 0x0000a004, 0x0000ffff
Local Switching: Enabled
Extended Counter: ON
Multicast Snooping: Disabled
Cluster-peer: enabled, Role:Active State: VPLS_MCT_STATE_OPER
```

Syntax: show mpls vpls detail

Use the **show mpls vpls brief redundancy** command to display PW redundancy.

Syntax: show mpls vpls brief redundancy

TABLE 78 Output from the show mpls vpls brief redundancy **command**

Field	Description
Name	The configured name of the VPLS instance.
Id	The ID of this VPLS instance.
Ports Up	The number of ports in this VPLS instance that are up.
Num Peers	The number of VPLS peers this device has for this VPLS instance.
Peers Up	The number of VPLS peers with which a VC connection is completely operational.
MCT PW- Role	Active: Node will start peering with remote peers, signaling Status TLV as Active. Standby: Node will start peering with remote peers , signaling Status TLV as Standby Transit: MCT VPLS FSM is not in Operation state. Remote Peering is not yet enabled.

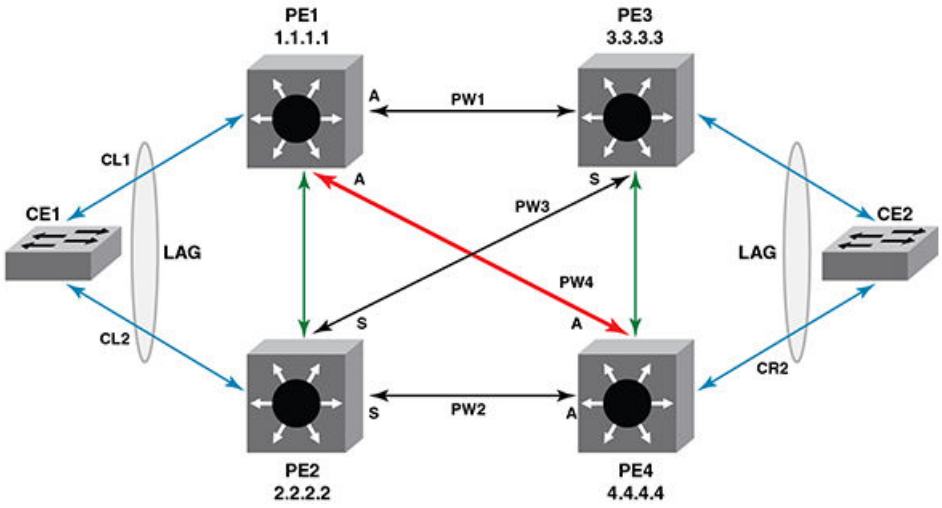
MCT for VLL

This feature supports dual-homing connectivity of CE devices to PE devices. Dual-homing enables link level and node level redundancy for CE's connected to PE's.

Figure 177 is a typical topology with MCT end-points for VLL. The VLL end-point connections from a single CE are dual homed to two PE nodes acting as an MCT cluster. From the CE, it has a LAG connection between CE and PE and is unaware of the MCT.

- PE1 and PE2 are two nodes of MCT cluster and work like a single PE. Similarly PE3 & PE4 are MCT cluster.
- CE1 is connected to PE1 and PE2 through a LAG.
- CE2 is connected to PE3 and PE2 through a LAG.
- PE1 has PW1 and PW3 vll PWs for same instance connected to remote MCT pairs.
- PE2 has PW2 and PW4 vll PWs for same instance connected to remote MCT pairs.
- For MCT pair PE1,PE2 PE1 is selected as active for a VLL instance, which it specifies as Active(A) in pw-redundancy status TLV during PW connection setup to remote PE pair.
- For MCT pair PE3 and PE4 nodes, PE4 is selected as active, which it specifies as Active(A) in pw-redundancy status TLV during PW connection setup to remote PE pair.
- There is a VLL instance mct-spoke-pw established between MCT pair for sending data traffic from standby to active node from the customer edge nodes(CE1, CE2).
- MCT cluster pair nodes communicate to each other using CCP communication provided by MCT infrastructure to identify as cluster pairs based on per instance and negotiate active/standby roles.
- For cluster pair PE1 and PE2, traffic from CE1 to CE2 flows as follows.
  - The traffic received directly from CE1 to PE1 is sent over the active PW PW3 to node PE4 which forwards it to client CE2.
  - The traffic received directly from CE1 to PE2 is sent over the MCT-SPOKE-PW to PE1. PE1 forwards this traffic over the active PW PW3 to node PE4 which forwards it to client CE2.
- For cluster pair PE1 and PE2, traffic from CE2 to CE1 flows as follows:
  - Traffic is received on the active PW for the pair PW3 on node PE1. Node PE1 forwards the traffic to CE1 directly over the local endpoint.
  - Traffic forwarded to PE2 in special case (where PE1 is Active but local CCEP is down)

FIGURE 174 Sample topology with MCT end-points for VLL



Configuration synchronization between MCT peers

MCT peers will exchange VLL information and updates.

VLL information sync

VLL addition and deletion information will be sent to MCT peer in the below scenarios.

- Whenever End Point (CCEP for MCT Client) and at least one VLL peer is configured, the VLL related information will sync to MCT peer. The MCT peer will add received VLL information to separate data structure (different from normal VLL structure).
- Whenever End Point (CCEP for MCT Client) or last VLL Peer are deleted then we will send message to MCT Peer to delete that VLL information.

Peer information sync

If VLL information is already sent to the MCT peer and VLL peer is updated, then the VLL peer information will sync to the MCT peer. VLL Peer information includes PW status and redundancy status.

End point status handling

Whenever there are any end point status changes, they are synchronized by MCT infrastructure and the events will be handled whenever logical status of endpoint changes from UP to DOWN or DOWN to UP.

End point mismatch

When end points configured in MCT nodes belongs to different MCT Clients, then the end point mismatch in both MCT nodes for that VLL and PWs for that VLL will be down.

Hitless upgrade

MCT VLL does not support Hitless upgrade, but is Hitless compatible.

Configuration Considerations

Brocade recommends making sure the VLL instance configuration is same on both nodes in the following aspects:

- Endpoint type(untagged or tagged or dual-tagged) and VLAN
- VC type(tagged-mode or raw-mode)
- Mtu and other operational parameters
- Peers configured for a given VLL instance

Configuring MCT VLL

MCT VLL requires following cluster level configurations. See Configuring the cluster operation mode on page 582 for additioan information on creating clusters.

L2VPN peer configuration

For each MCT VLL instance, a spoke-PW session is formed internally using the **l2vpn-peer** command to form a PW session to the remote MCT node. In this example, spoke-PW will be formed to peer 2.2.2.2.

```
device(config-mpls)#l2vpn-peer 2.2.2.2 rbridge-id 2
```

Syntax: [no] l2vpn-peer *ip-address* **rbridge-id** *id*

The *ip-address* variable specifies the IP address of the targeted peer.

The *id* parameters specify the remote bridge ID. Possible values are 1 - 35535 (16 bit value).

To disable the configuration, enter the **no** form of the command.

VLL global pw-redundancy (optional)

Once MCT is enabled for a VLL instance, the two MCT cluster nodes synchronize the configuration with each other over the CP and decide which node will take up the Active role and Standby role. PW redundancy support provides is backup PWs ready so that traffic can be quickly failed over to the backup PWs. This command can be configured either globally for all VLL instances with MCT or for each VLL instance individually.

> **NOTE**
> If it is not configured, the MCT node with lower rbridge-id will be elected as Active to signal to the remote PE's.

```
Use the vll-pw-redundancy-active
 command at the global mode to set the pw-redundancy option for all VLL Instances with MCT.
Brocade(config-cluster-PE1-client-2)# rbridge-id 101
Brocade(config-cluster-PE1-client-2)# client-interface ethernet 1/1
Brocade(config-cluster-PE1-client-2)# vll-pw-redundancy-active
Brocade(config-cluster-PE1-client-2)# deploy
```

Syntax: [no] vll-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

Per VLL instance pw-redundancy (optional)

If **vll-pw-reduncancy-active** is not configured per VLL instance, the selection will be based on the global configuration. The per VLL Instance configuration always has the higher priority over global configuration.

Vll-pw-redundancy-active option is used to load-balance VLL instances using client (for one client VLLs, one node is active and for other client, another MCT node can be active).

All VLL instances will not be active only on one node. This allows flexibility to provision VLL instances for each client differently.

When the client is deployed, all VLL instances that share the same MCT end-point(port e 1/1 in the example) will become MCT VLLs.

When the client configuration is un-deployed the VLL instance will be brought down and up without MCT.

The **vll-pw-redundancy-active** option if configured cannot be changed without un-deploying the client.

For VLL, two remote peers (remote MCT cluster pair) must be configured to support PW redundancy. In the configuration below, 3.3.3.3 and 4.4.4.4 are peers to reach remote node.

```
device(config-mpls)#vll test 10
device(config-mpls-vll-test)#vll-peer 3.3.3.3 [4.4.4.4]
device(config-mpls-vll-test)#vll-pw-redundancy-active
```

Syntax: [no] vll-pw-redundancy-active

The **no** form of this command removes the pw redundancy option.

If only one peer is specified, the PW redundancy status TLV with local status as active is supported.

Once the vll-instance is operational and if the vll level pwredundancy is changed, the pw-redundancy election is triggered, which can cause the vll-active state to change.

Setting the L2VPN global revertible timer

Whenever a node needs to become active for a given L2VPN instance, it sends a message to peer MCT node and starts this timer. Once this timer expires on configured active MCT node, it will move the L2VPN sessions to be active and the remote peer MCT node will move the sessions to be standby state for PW operation.

Use the **l2vpn-revertible-timer** command to start a revertible timer whenever a given L2VPN instance moves to different node as active node (against the configured active).

```
device(config-mpls)# l2vpn-revertible-timer 200
```

Syntax: [no] l2vpn-revertible-timer *sec*

Use the *sec* parameter to specify the amount to time before the L2VPN session becomes active. The default time is 300 seconds and can be a value from 0 throgh 65535 seconds. If the value is configured to be 0, then reversion will happen immediately.

> NOTE
> Immediate reversion can cause instability.

PW redundancy auto reversion timer option

If there is a transient condition where the standby node's pw is active and active node pw comes up, an auto-reversion timer of 60 seconds value(eg: 60) +/- 25% jitter seconds is started on the active node. Upon the auto reversion timer expiration, the remote pw on the active node changes the role to active and the remote pw on the standby node changes role to standby. This role change is coordinated between the MCT active and standby nodes.

Display commands

Use the **show mpls vll brief** and **show mpls detail** commands to display PW redundancy information for all VLLs and MCT related information.

```
Brocade#show mpls vll brief
* - Active VLL Peer; U - UP; D - DOWN
                                                          Vll-Peer          Vll-Peer          MCT
Name      VC-ID       End-Point                           (State)           (State)           state
```

```
----      -----       ---------                              --------          --------          -----
vl            1        untag e 1/4(U)                         5.5.5.5(U)*       4.4.4.4(U)     Active
v2            2        tag vlan 201 e 1/4(U)                  4.4.4.4(U)*       5.5.5.5(U)     Active
```

Syntax: show mpls vll brief

```
Brocade# show mpls vll detail
VLL VLL1, VC-ID 1, VLL-INDEX 0
  End-point        : untagged   e 1/4
  End-Point state  : Up
  MCT state        : Active
  Local VC type    : tag
  Local VC MTU     : 4974
  COS              : --
  Extended Counters: Disabled
  Counter          : Disabled
  Vll-Peer         : 5.5.5.5 (Standby-Standby)
    State          : Down - no tummel LSP to vll-peer
    Remote VC type :                Remote VC MTU  :
    Local label    :                   Remote label   :
    Local group-id : 0                 Remote group-id:
    Tunnel LSP     : LSP_to_1 (tnl0)
    MCT Status TLV : Standby
  Vll-Peer         : 4.4.4.4 (Active-Active)
    State          : UP
    Remote VC type : tag               Remote VC MTU  : 1500
    Local label    : 798720            Remote label   : 798722
    Local group-id : 0                 Remote group-id: 0
    Tunnel LSP     : LSP4 (tnl2)
    MCT Status TLV : Active
  MCT Infomation   :
    Local CCEP state      : UP
    Remote CCEP state     : Down
    Pending reversion time: --
    Spoke PW         : 1.1.1.1
      State          : DOwn - endpoint is not UP (Reason: Election not done)
      Remote VC type :                   Remote VC MTU  :
      Local label    :                   Remote label   :
      Local group-id :                   Remote group-id: 0
      Tunnel LSP     : LSP_to_5 (tnl2)
    MCT Peer PWs     :
      VLL Peer           UP/DOWN  Active/Standby
      --------           -------  --------------
      1.1.1.1            UP       Active-Standby
      10.10.10.10        UP       Active-Active
```

Syntax: show mpls vll detail

MCT VLL sample configuration

```
Brocade# show running configuration
router mpls
  l2vpn-revertible-timer 300
  lsp MCT_PEER_LSP            /
    to 2.2.2.2
    enable
  vll MCT_VLL1 101
     vll-peer 3.3.3.3 [4.4.4.4]
     [vll-pw-redundancy-active]
     vlan 101
       tagged eth 1/1
cluster mct_l2vpn 1
     rbridge-id 1
     l2vpn-peer 2.2.2.2 rbridge-id 2
    deploy
    client MCT_CLIENT1
      rbridge-id 101
      client-interface e 1/1
         [vll-pw-redundancy-active]
      deploy
```

MCT Snooping

All VLAN IGMP and PIM snooping features are supported on the MCT links. Brocade supports both IPv4 and IPv6 snooping over MCT links.

Events Handling

CCEP down event to MCT

CCEP down event to MCT peer is generated only if,

- The CCEP port is replaced in the SG entry's OIF to ICL ports, if CCEP ports exists in the OIF list. The new periodic IGMP joins now coming through the remote CCEP ports will reach us as MDUP messages on the ICL ports. This will ensure that the ICL ports in the SG/WG OIF list and IGMPv3 DB are refreshed.

MCT remote CCEP down event

MCT remote CCEP down event is generated only if,

The CCEP ports of the MCT peer (remote CCEP ports) goes down. Two possible scenarios exist:

- SG entry with ICL as IIF already exists (because there was a local CEP receiver for that SG). In this case, a CCEP link (only if it exists in WG entry or IGMPv3 DB due to joins received) will be added to SG OIF list. If no joins were received from CCEP, then the CCEP will not be added to the OIF list, unless it is a flooding scenario.
- SG entry with ICL as IIF did not exist because there are only remote CCEP receivers and no local CEP receivers. In this case when remote CCEP goes down we'll start getting SG flows through ICL and we'll create SG and add local CCEP ports to the OIF (copied from WG entry).

MCT local CCEP up event

MCT local CCEP up event is generated only:

After the local CCEP comes up, two possible scenarios exist:

- IGMP joins may start coming through the local CCEP links or
- IGMP joins may continue to come through the remote CCEP link as before.

Case 1: You stop receiving MDUP message from the MCT peer through the ICL link and the ICL ports are eventually aged out from the WG/SG OIF list. Local CCEP ports are added to the OIF list.

Case 2: You will continue to get MDUP messages from MCT peer through the ICL link and the local CCEP is up and the local CCEP is added to the OIF. The ICL ports will eventually age out, if the MDUP join messages with remote CEP flag set were not received. The CCEP up event is sent to the MCT PEER.

MCT remote CCEP up event

CCEP up event from the MCT PEER is generated only if,

- IGMP joins may start coming through the remote CCEP links or
- IGMP joins may continue to come through the local CCEP link as before.

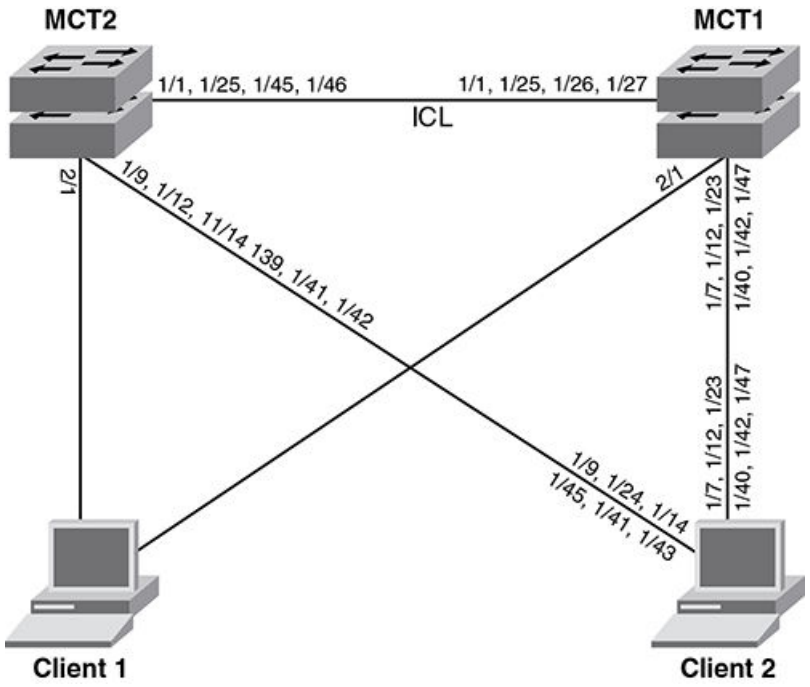
In both cases, upon receiving MCT CCEP up event from peer, the local CCEP ports will be removed from the OIF list of those SG entries with ICL as IIF. No other SG entries are affected. They'll continue to have the membership information from CCEP ports.

In the second case, it will continue to send MDUP join messages (with CCEP flag set) to the MCT peer, as usual. This will ensure that, in the MCT peer, those SG's sourced from its CEP ports will have its CCEP links as OIF (efficient path).

Configuration considerations

- IGMP and PIM proxy configurations need to be configured on both the MCT peers.
- Both MCT nodes should have same snooping mode (either both should be active or both should be passive).
- Static IGMP membership configurations for CCEP ports need to configured on both the MCT peers.
- Static uplink configurations need to be configured on both the MCT peers.
- MDUP does not support fragmentation of control packets over ICL.
- The IP MTU of any non-Brocade PIM router connected to a Brocade MCT PIM Snooping node needs to be configured to <=1450 bytes. This is due to the CCP payload size limit.

FIGURE 175 MCT snooping topology example



There are no specfic MCT commands for configuring MCT snooping. Below is a sample MCT snooping configuration .

Sample MCT snooping configuration

```
!
lag "CLIENT-1" dynamic id 6
ports ethernet 2/1
 primary-port 2/1
deploy
!
!
lag "CLIENT-2" static id 5
ports ethernet 1/7 ethernet 1/12 ethernet 1/23 ethernet 1/40 ethernet 1/42 ethernet 1/47
```

```
 primary-port 1/47
deploy
!
!
lag "ICL-1" static id 4
ports ethernet 1/1 ethernet 1/25 to 1/27
 primary-port 1/1
deploy
!
!
vlan 21
 untagged ethe 1/7 ethe 1/12 ethe 1/23 ethe 1/40 ethe 1/42 ethe 1/47 ethe 2/1
 tagged ethe 1/1 ethe 1/25 to 1/27
router-interface ve 21
multicast active
!
vlan 31
 tagged ethe 1/1 ethe 1/7 ethe 1/12 ethe 1/23 ethe 1/25 to 1/27 ethe 1/40 ethe 1/42 ethe 1/47 ethe 2/1
router-interface ve 31
multicast6 passive
!
!
vlan 4090
 tagged ethe 1/1 ethe 1/25 to 1/27
 router-interface ve 49
!
interface ve 21
ip address 21.1.1.1/24
!
interface ve 31
ip address 31.31.31.3/24
ipv6 address 2031:31::2/112
!
interface ve 49
ip address 49.49.49.1/30
!
!
cluster CLUSTER-1 1
 rbridge-id 1
 session-vlan 4090
 member-vlan 20 to 50
icl ICL-1 ethernet 1/1
peer 49.49.49.2 rbridge-id 35535 icl ICL-1
deploy
 client CLIENT-1
  rbridge-id 3
  client-interface ethernet 2/1
  deploy
 client CLIENT-2
  rbridge-id 2
  client-interface ethernet 1/47
  deploy
!
Config on MCT2:-
=====================
!
lag "CLIENT-1" dynamic id 6
ports ethernet 2/1
 primary-port 2/1
deploy
!
lag "CLIENT-2" static id 5
ports ethernet 1/9 ethernet 1/12 ethernet 1/14 ethernet 1/39 ethernet 1/41 to 1/42
 primary-port 1/39
deploy
!
!
lag "ICL-1" static id 1
ports ethernet 1/1 ethernet 1/25 ethernet 1/45 to 1/46
 primary-port 1/1
deploy
!
```

```
!
vlan 21
 untagged ethe 1/9 ethe 1/12 ethe 1/14 ethe 1/39 ethe 1/41 to 1/42 ethe 2/1
 tagged ethe 1/1 ethe 1/25 ethe 1/45 to 1/46
router-interface ve 21
multicast active
!
!
vlan 31
 tagged ethe 1/1 ethe 1/9 ethe 1/12 ethe 1/14 ethe 1/25 ethe 1/39 ethe 1/41 to 1/42 ethe 1/45 to 1/46 ethe
2/1
 router-interface ve 31
multicast6 passive
!
!
vlan 4090
 tagged ethe 1/1 ethe 1/25 ethe 1/45 to 1/46
 router-interface ve 49
!
interface ve 21
ip address 21.1.1.2/24
!
interface ve 31
ip address 31.31.31.2/24
ipv6 address 2031:31::1/112
!
!
interface ve 49
ip address 49.49.49.2/30
!
!cluster CLUSTER-1 1
 rbridge-id 35535
 session-vlan 4090
 member-vlan 20 to 50
icl ICL-1 ethernet 1/1
peer 49.49.49.1 rbridge-id 1 icl ICL-1
deploy
 client CLIENT-1
  rbridge-id 3
  client-interface ethernet 2/1
  deploy
 client CLIENT-2
  rbridge-id 2
  client-interface ethernet 1/39
  deploy
```

Displaying IP multicast information

The following sections show how to display IP multicast information.

Displaying multicast information

The existing show commands have been modified to include the MCT specific information in the outgoing interfaces.

The example below displays if IGMP/MLD snooping is enabled on VLAN 20 and what mode of snooping is configured (active/passive).

Cluster peer #1:

```
device# show ip multicast vlan 20
----+-----+---------+---------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+---------------+-----+-----+------
20   I-Ena Active    Self            40    1     1
----+-----+---------+---------------+-----+-----+------
Router ports:
Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join
  1    (*, 224.1.1.1 ) 00:09:04 NumOIF: 2 profile: none
```

```
          Outgoing Interfaces:
               CCEP rbr-id 200 vlan 20 ( V2) 00:00:22/22s
               ICL  vlan 20 ( V2) 00:07:37/16s
  1    (1.1.1.1, 224.1.1.1) in e2/1 vlan 20 00:00:03 NumOIF: 2 profile: none
          Outgoing Interfaces:
               ICL e2/15 vlan 20 ( V2) 00:00:03/0s
               CCEP rbr-id 200 e2/7 vlan 20 ( V2) 00:00:03/0s
          FID: 0x8006 MVID: None
```

Cluster Peer 2:

```
DUT2#show ip multicast vlan 20
----+-----+---------+---------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+---------------+-----+-----+------
20   I-Ena Passive   None            28    1     1
----+-----+---------+---------------+-----+-----+------
Router ports:
Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join
  1    (*, 224.1.1.1 ) 00:09:24 NumOIF: 2 profile: none
          Outgoing Interfaces:
               CCEP rbr-id 200 vlan 20 ( V2) 00:00:42/42s
               e2/2 vlan 20 ( V2) 00:05:15/36s
  1    (1.1.1.1, 224.1.1.1) in e2/15 vlan 20 00:00:23 NumOIF: 1 profile: none
          Outgoing Interfaces:
               e2/2 vlan 20 ( V2) 00:00:23/0s
          FID: 0x8006 MVID: None
```

MCT Client:

```
device# show ip multicast vlan 20
----+-----+---------+---------------+-----+-----+------
VLAN State Mode      Active          Time (*, G)(S, G)
                     Querier         Query Count Count
----+-----+---------+---------------+-----+-----+------
20   I-Ena Passive   77.77.77.1      81    1     1
----+-----+---------+---------------+-----+-----+------
Router ports: 4/3 (97s)
Flags: R-Router Port, V2|V3: IGMP Receiver, P_G|P_SG: PIM Join
  1    (*, 224.1.1.1 ) 00:07:09 NumOIF: 2 profile: none
          Outgoing Interfaces:
               e4/3 vlan 20 ( R) 00:02:42/97s
               e3/2 vlan 20 ( V2) 00:07:09/91s
  1    (1.1.1.1, 224.1.1.1) in e4/3 vlan 20 00:02:20 NumOIF: 1 profile: none
          Outgoing Interfaces:
               e3/2 vlan 20 ( V2) 00:02:20/0s
          FID: 0x8005 MVID: None
```

Syntax: show ip multicast vlan *vlan-id*

The **vlan** *vlan-id* parameter displays IP multicast VLAN information for a specified VLAN.

Displaying IP multicast statistics

To display IP multicast statistics on a device, enter the following commands at any level of the CLI.

```
device##show ip multicast vlan 20 statistics
VLAN ID 20
  Receive stats:                        Transmit stats:
  General query          : 0            General query          : 3
  Group specific query   : 0            Group specific query   : 0
  IGMP Report            : 14           IGMP V2 Proxy Sent     : 0
  IGMP Leave             : 1            IGMP V3 Proxy Sent     : 0
  IGMPV3 Report          : 0            PIM Proxy Sent         : 0
  IGMPV3 Error           : 0            MCT MDUP msg recvd     : 10
  PIMV2 hello            : 0
  PIMV2 join/prune       : 0
  PIMV2 J/P pkt error    : 0
```

```
MCT MDUP msg sent       : 5
MCT MDUP msg error      : 0
```

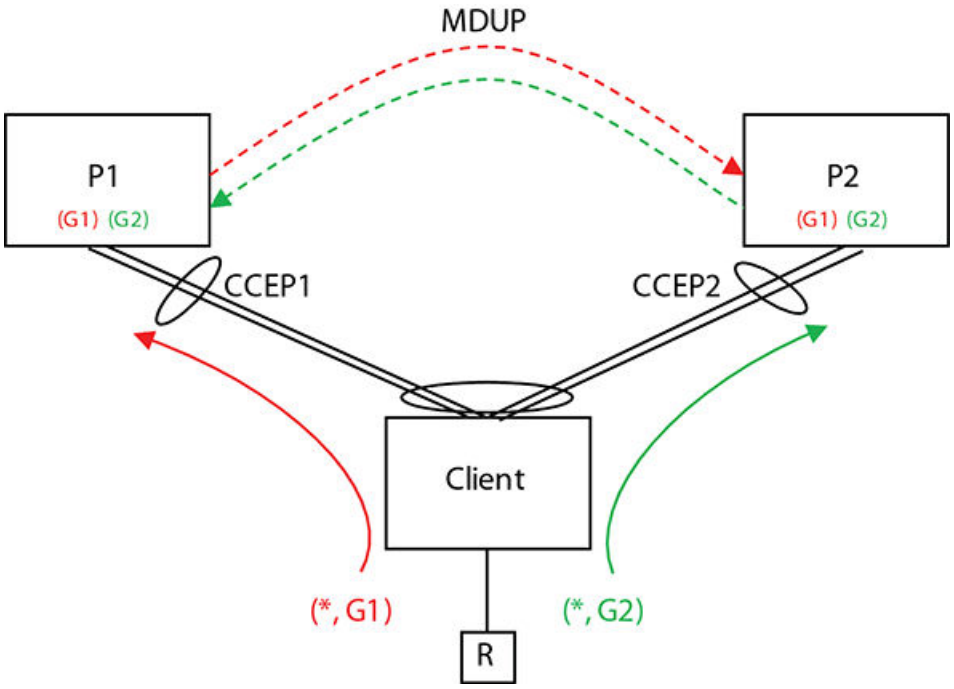
Syntax: show ip multicast statistics

For information regarding the output fields, refer to "Displaying IP multicast information".

PIM Over MCT

Figure 179 is an example of the IGMP synchronization.

FIGURE 176 Synchronizing IGMP State on the CCEPs



Synchronizing IGMP State on the CCEPs

The MDUP channel sends an IGMP report of messages received on the local CCEP to the remote MCT peer. This ensures that regardless of which MCT peer the client forwards the messages, both will recieve it and process it, ensuring identical state in both MCT Peers.

PIM Over MCT on page 650 displays R sending IGMP report for (*, G1).

NOTE

If the multicast source lies on the uplink, and receivers lie on the MCTclients, both MCT nodes will forward traffic to the clients if the ICL goes down. To avoid this situation, the keepalive VLAN must always be configured under the cluster to enable Master-Slave election and prevent duplicate traffic to CCEP receivers.

Synchronization method.

Client forwards it to P1 on CCEP1.

P1 sends this to P2 via MDUP.

P2 adds (*, G1) to CCEP2 upon processing it.

R sends IGMP report for (*, G2).

Client forwards it to P2 on CCEP2.

P2 sends this to P1 via MDUP.

P1 adds (*, G2) to CCEP1 upon processing it.

Final result, both P1 and P2 has both (*, G1) and (*, G2) on their local CCEPs.

Traffic Load sharing on the CCEPs

For a stream that has a receiver behind the Client, both MCT Peers recieve the traffic, and each has to decide whether to forward the traffic to the local CCEP or let the remote peer take care of it. The following simple decision function is used to determine this.

- If the CCP is down, it will forward locally.
- If the remote CCEP is down, it will forward locally.
- If the local CCEP is down, it will not forward locally.
- If the ingress is the CEP, it will forward locally.
- If the ingress is the ICL, it will not forward locally.
- If the ingress is a different CCEP, it will forward locally.

Use the following expression to determine the traffic load sharing.

Forward locally =

((Src_addr + Grp_addr) & 0x00000001) ^ ((UINT32)(local_brdige_id > remote_bridge_id))

Sending IGMP Queries on CCEPs

Since there are two chassis connected to the same MCT VLAN and since the Client is not going to flood the incoming IGMP Queries to the other chassis, both chassis could end up electing themselves as the IGMP Querier and sending queries on the CCEPs and the hosts behind the client will end up responding to both. To avoid this, the IGMP surpresses queries going out of the CCEP port on one of the MCT Peers. The following algorithm is used:

If (CCP connection is down)

Send Queries on local CCEP.

Else if (remote CCEP is down)

Send Queries on local CCEP.

Else if (local CCEP is down)

Suppress Queries on local CCEP.

Else if (local-bridge-id > remote-bridge-id)

Send Queries on local CCEP.

Else

Suppress Queries on local CCEP.

Show commands

Use the **show ip pim mcache** command to display if a CCEP port is being blocked or being forwarded to.

```
device#show ip pim mcache
IP Multicast Mcache Table
Entry Flags    : SM  - Sparse Mode, SSM - Source Specific Mutlicast, DM - Dense Mode
                 RPT    - RPT Bit, SPT - SPT Bit, LSRC - Local Source, LRCV - Local Receiver, JOIN - Join
Upstream
                 HW - HW Forwarding Enabled, FAST - Resource Allocated, TAG - Need For Replication Entry
                 REGPROB - Register In Progress, REGSUPP - Register Suppression Timer
                 MSDPADV - Advertise MSDP, NEEDRTE - Route Required for Src/RP,  PRUN - DM Prune Upstream
Interface Flags: IM - Immediate, IH - Inherited, WA - Won Assert
                 MJ - Membership Join, MI - Membership Include, ME - Membership Exclude
                 BR - Blocked RPT, BA - Blocked Assert, BF - Blocked Filter, BI - Blocked IIF,
            BM - Blocked MCT
Total entries in mcache: 1
1     (2.2.2.101, 239.0.1.3) in v200 (e2/15), Uptime 00:01:08, Rate 42229 (SM)
      Source is directly connected. RP 2.2.2.1
      Flags (0x3046cec1) SM SPT L2REG LSRC LRCV JOIN HW FAST MSDPADV
      fast ports: ethe 2/1
      AgeSltMsk: 00000002,  FID: 0x8006, MVID: NotReq, RegPkt: 0, AvgRate: 41688, profile: none
      Forwarding_oif: 1, Immediate_oif: 1, Blocked_oif: 1
      L2 (HW) 1:
          TR(e2/1,e2/1),  00:01:08/181, Flags: IM IH
      Blocked OIF 1:
          TR(e1/5,e1/5)(VL200), 00:01:08/0, Flags: MJ BM
Number of matching entries: 1
device#
```

Syntax: show ip pim mcache

Refer to "Displaying the PIM multicast cache" for output descriptions.

Use the **show ipv6 pim mcache** command to display if a CCEP port is being blocked or being forwarded to.

```
MLX#show ipv6 pim mcache
IP Multicast Mcache Table
Entry Flags    : SM  - Sparse Mode, SSM - Source Specific Mutlicast, DM - Dense Mode
                 RPT    - RPT Bit, SPT - SPT Bit, LSRC - Local Source, LRCV - Local Receiver, JOIN - Join
Upstream
                 HW - HW Forwarding Enabled, FAST - Resource Allocated, TAG - Need For Replication Entry
                 REGPROB - Register In Progress, REGSUPP - Register Suppression Timer
                 MSDPADV - Advertise MSDP, NEEDRTE - Route Required for Src/RP,  PRUN - DM Prune Upstream
Interface Flags: IM - Immediate, IH - Inherited, WA - Won Assert
                 MJ - Membership Join, MI - Membership Include, ME - Membership Exclude
                 BR - Blocked RPT, BA - Blocked Assert, BF - Blocked Filter, BI - Blocked IIF,
            BM - Blocked MCT
Total entries in mcache: 1
1 (2062:62:62:62::11, ff1a::2) in v62 (tag e1/1), Uptime 00:00:58 (SM)      upstream neighbor is L2
fe80::21b:edff:fea4:a441. RP 2001:1:2:3:4::b      Flags (0x304680c1) SM SPT LSRC LRCV HW FAST      fast
ports:       AgeSltMsk: 00000003, FID: 0xffff (D), DIT: NotReq, profile: none, KAT Timer value:
240      Forwarding_oif: 0, Immediate_oif: 0, Blocked_oif: 1       Blocked OIF 1:
            TR(e1/39,e1/39)(VL62), 00:00:01/0, Flags: MJ BM
Number of matching entries: 1
MLX#
```

Refer to "Displaying the IPv6 PIM multicast cache" for output descriptions.

Displaying MCT PIM Count

To display the statistics and error counters for the Multicast MDUP channel between the MCT peers, use the **show ip pim count mct** command.

```
device#show ip pim count mct
Multicast MCT Statistics for IPv4 (UP):
Messages assembled into the send buffer  : 11811
Messages processed out of the recv buffer: 0
```

```
Segments sent successfully to TCP        : 11762
Segments failed to be accepted by TCP    : 0
Segments assembled into the receive buffer : 0
Messages dropped because (size > 1500)     : 0
Messages dropped because it won't fit into available space in send buffer   : 0
Segments dropped because it won't fit into available space in receive buffer: 0
Received messages dropped because of cluster-id mismatch                    : 0
Received messages dropped because the peer was not recognized               : 0
Received messages dropped because cluster not active                        : 0
Received messages dropped because MCT VLAN unrecognized                     : 0
Received messages dropped because of bad message type                       : 0
Received messages dropped because of bad checksum                           : 0
Received bytes skipped because of sync or checksum errors                   : 0
device#
```

Syntax: show ip pim count mct

To display the MCT IPv6 PIM count, use the **show ipv6 pim count mct** command.

```
device#show ipv6 pim count mct
Multicast MCT Statistics for IPv6 (UP):
Messages assembled into the send buffer  : 279
Messages processed out of the recv buffer: 523
Segments sent successfully to TCP        : 279
Segments failed to be accepted by TCP    : 0
Segments assembled into the receive buffer : 293
Messages dropped because (size > 1500)     : 0
Messages dropped because it won't fit into available space in send buffer   : 0
Segments dropped because it won't fit into available space in receive buffer: 0
Received messages dropped because of cluster-id mismatch                    : 0
Received messages dropped because the peer was not recognized               : 0
Received messages dropped because cluster not active                        : 0
Received messages dropped because MCT VLAN unrecognized                     : 0
Received messages dropped because of bad message type                       : 0
Received messages dropped because of bad checksum                           : 0
Received bytes skipped because of sync or checksum errors                   : 0
device#
```

Syntax: show ipv6 pim count mct

Displaying IGMP Interfaces

Use the **show ip igmp interface** command to show the IGMP Query suppression state on CCEP ports.

```
device#show ip igmp interface
---------+------+---------+---------------+---------+-----+-----+---------
Intf/Port|Groups| Version |Querier        |  Timer  |V1Rtr|V2Rtr|Tracking
         |      |Oper  Cfg|               |OQrr GenQ|     |     |
---------+------+----+----+---------------+----+----+-----+-----+---------
v200           2    2    -                                        Disabled
    e2/15           2    - Self               0   59 No    No
     e2/1           2    - Self               0   59 No    Yes
     e1/5           2    - Self (MCT-Blk)     0   40 No    No
device#
```

Refer to "Displaying the IGMP status of an interface" for output descriptions.

Displaying MLD Interfaces

Use the **show ipv6 mld interface** command to show the IGMP Query suppression state on CCEP ports.

```
device#show ipv6 mld interface
---------+------+---------+-----------------------------------------+---------+-----+---------
Intf/Port|Groups| Version |Querier                                  |  Timer  |V1Rtr|Tracking
         |      |Oper  Cfg|                                         |OQrr GenQ|     |
---------+------+----+----+-----------------------------------------+----+----+-----+---------
v62            0    2    2                                                      Disabled
     e2/1           2    - Self (MCT-Blk)                             0   79 No
    e1/37           2    - Self                                       0  108 No
```

```
    e1/33           2    - Self                                    0  108 No
device#
```

Refer to "Displaying IPv6 PIM interface information" for output descriptions.

Sample configuration

MCT Peer 1

```
lag "AMERICAS-CCEP-1-1" static id 11
 ports ethernet 1/5 to 1/6
 primary-port 1/5
 deploy
lag "AMERICAS-ICL" static id 1
 ports ethernet 2/1 to 2/2
 primary-port 2/1
 deploy
vlan 200
 untagged ethe 2/15
 tagged ethe 1/5 to 1/6 ethe 2/1 to 2/2
 router-interface ve 200
vlan 4090
 tagged ethe 2/1 to 2/2
 router-interface ve 100
router pim
 rp-address 2.2.2.1
interface management 1
 ip address 10.25.109.6/21
 enable
interface ve 100
 ip address 1.1.1.1/24
interface ve 200
 ip address 2.2.2.1/24
 ip pim-sparse
cluster AMERICAS 1
 rbridge-id 100
 session-vlan 4090
 member-vlan 200
 icl AMERICAS-ICL ethernet 2/1
 peer 1.1.1.2 rbridge-id 200 icl AMERICAS-ICL
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client AMERICAS-CLIENT-1
  rbridge-id 300
  client-interface ethernet 1/5
  deploy
```

MCT Peer 2

```
lag "AMERICAS-CCEP-2-1" static id 21
 ports ethernet 3/5 to 3/6
 primary-port 3/5
 deploy
lag "AMERICAS-ICL" static id 1
 ports ethernet 4/1 to 4/2
 primary-port 4/1
 deploy
vlan 200
 untagged ethe 4/23
 tagged ethe 3/5 to 3/6 ethe 4/1 to 4/2
 router-interface ve 200
vlan 4090
 tagged ethe 4/1 to 4/2
 router-interface ve 100
router pim
 rp-address 2.2.2.1
interface management 1
 ip address 10.25.109.5/21
```

```
 enable
interface ve 100
 ip address 1.1.1.2/24
interface ve 200
 ip address 2.2.2.2/24
 ip pim-sparse
cluster AMERICAS 1
 rbridge-id 200
 session-vlan 4090
 member-vlan 200
 icl AMERICAS-ICL ethernet 4/1
 peer 1.1.1.1 rbridge-id 100 icl AMERICAS-ICL
 client-interfaces sync_ccep_early lacp-delay 5
 deploy
 client AMERICAS-CLIENT-1
  rbridge-id 300
  client-interface ethernet 3/5
  deploy
```

BFD over MCT

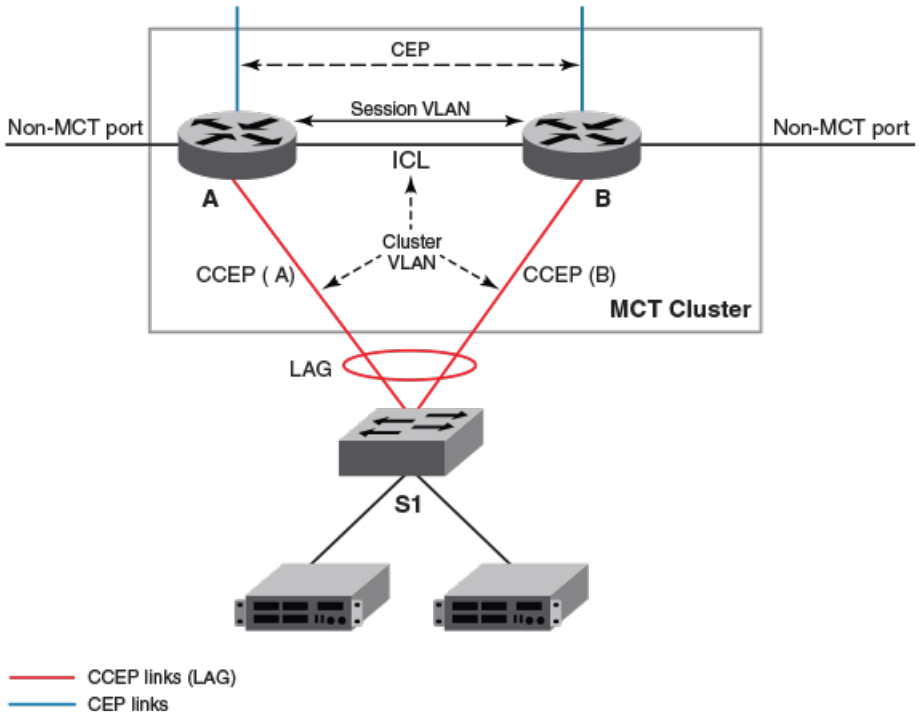
Bidirectional Forwarding Detection (BFD) over MCT detects forwarding path failures in different applications that are configured on MCT cluster devices.

BFD sessions are formed between clients of MCT clusters and MCT nodes. BFD sessions over an MCT member virtual ethernet (VE) interface detect forwarding path failures in the following applications:

- Static routes
- OSPFv2 and OSPFv3
- IS-IS
- BGP4 and BGP4+

For more information about BFD, refer to the *Bidirectional Forwarding Detection* chapter.

FIGURE 177 BFD over MCT



In the network diagram, applications on router A send requests to BFD to detect the status of the protocol peers on S1 and clients beyond S1. The following events are observed when BFD is implemented on router A.

BFD at the Rx port:

- Successfully receives periodic BFD packets sent by the MCT client and maintains the BFD state.
- The receiving port can be either ICL or CCEP.
- Detects the loss of connectivity with the application peer on the MCT client within BFD time constraints.
- Differentiates a CCEP or ICL failure from a complete breakdown.

BFD at the Tx port:

- Sends periodic BFD control packets to the MCT client directly via CCEP or via ICL ports.
- Moves BFD Tx to ICL before BFD times out on the MCT client when CCEP is down.
- Moves BFD Tx to CCEP before BFD times out on the MCT client when the ICL is down.

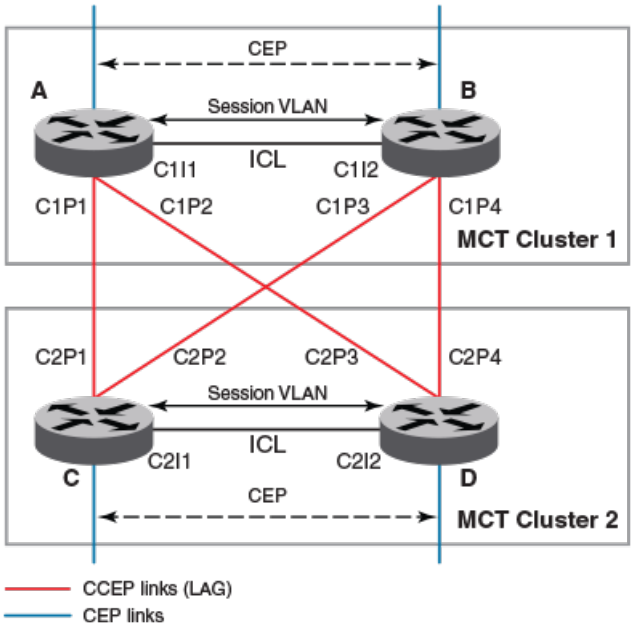
Use case: BFD over MCT with multiple LAGs

BFD sessions are formed between MCT cluster clients and MCT nodes. The VEs span ICL and CCEP ports; thus BFD packets from clients can reach MCT nodes on either the CCEP port or the ICL link. The local BFD Tx port can be on either of these ports. To support this configuration, BFD session processing is modified to accept and send packets on VEs with multiple LAGs as members.

Let us consider that a BFD session formed between A and C, as shown in the following network diagram. The following scenarios describe how the BFD Tx port is switched when a port goes down to ensure that there is no impact to the ongoing BFD session.

- Router A's Tx port is C1P1, and router C's Rx port is C2P1. When C1P1 is down, router A's Tx port switches to ICL port C1I1. When router C receives the BFD packet on port C2I1, it also switches its Rx port to the C2I1 port.
- Router A's Tx port is C1P2, and router C's Rx port is C2I1. When C1P2 goes down, router A's Tx port switches to ICL port C1I1, while router C's Rx port does not change.
- Router A's Tx port is C1I1, and router C's Rx port is C2P2. When C1I1 goes down, router A's Tx port changes to either the C1P1 port or the C1P2 port. When router C receives the BFD packet on C2P1 or C2I1, its Rx port also changes to C2P1 or C2I1, respectively.

FIGURE 178 BFD over MCT with multiple LAGs



BFD over MCT limitations

- BFD sessions may go down or flap when MAC learning occurs on MCT devices.
- BFD over MCT does not support VE over Virtual Private LAN Service (VPLS).

Upgrade and downgrade considerations

- During a network upgrade procedure, BFD sessions are established between the MCT client and the upgraded MCT device.
- BFD should be disabled on MCT member VEs before the downgrade to prevent impact on certain applications.

BFD over MCT scalability

- BFD over MCT supports all members of the LAG.
- BFD over MCT supports BFD timers for values between 100 and 30000 milliseconds.

NOTE
When Brocade NetIron CER or CES devices are heavily loaded with many BFD sessions, the BFD sessions may flap if the configured BFD interval is less than 500 ms with a multiplier value of 3. For multihop IPv4 session, BFD sessions are stable at 500 ms with a multiplier value of 3. For multihop IPv6 session, BFD sessions are stable at 600 ms with a multiplier value of 3.

- On the Brocade NetIron MLX and XMR devices, 250 BFD sessions are supported. However, 40 BFD sessions are supported on an LP.
- On the Brocade NetIron CES and CER devices, 40 BFD sessions are supported. 40 BFD sessions are supported on an LP.
- When BFD supports a maximum of 40 session per LP or a LAG, BFD sessions may switch to ICL LAG from CCEP LAG when CCEP is down or an interface in CCEP LAG is down. Hence BFD can support a maximum of 40 sessions over MCT cluster.
- For the configuration tabulated below on a Brocade NetIron CES or CER device, BFD is stable at a timer value of 5000 msec * 3. At timer value lesser than 5000 msec * 3, BFD session may flap.

Scenario or protocol	Session type	Session type
OSPF	**IPv4 session**	**IPv6 session**
Neighbor	5	5
Routes	130K	1.5K
BGP	**IPv4 session**	**IPv6 session**
Neighbor	5	5
Routes	100K	4K
MPLS	**Configured**	**Peer**
VLL	100	300
VPLS	256	768
Number of MPLS tunnels allocated	30	
Number of MPLS cross-connects allocated	56	
BFD		
Number of BFD sessions	20	

Configuring BFD over MCT

BFD over MCT is configured to detect forwarding path failures in different applications that are configured on MCT cluster devices.

1. Configure the MCT cluster on the Brocade device.
   a) Create LAG for CCEP ports

```
device(config-lag-ccep)#
device(config-lag-ccep)# lag ccep dynamic id 2
device(config-lag-ccep)# ports ethernet 4/17 ethernet 4/18
device(config-lag-ccep)# primary-port 4/17
device(config-lag-ccep)# deploy
```

   b) Create LAG for ICL ports

```
device(config)# lag icl dynamic id 1
device(config-lag-icl)# ports ethernet 1/2 ethernet 3/3
device(config-lag-icl)# primary-port 1/2
device(config-lag-icl)# deploy
```

c) Configure member VLAN

```
device(config)# vlan 7 name member_vlan
device(config-vlan-7)# tagged ethernet 1/2 ethernet 3/3 ethernet 4/17 ethernet 4/18
device(config-vlan-7)# router-interface ve 7
```

d) Configure session VLAN

```
device(config)# vlan 10 name session_vlan
device(config-vlan-10)# tagged ethernet 1/2 ethernet 3/3
device(config-vlan-10)# router-interface ve 10
```

e) Assign IP address to session VLAN virtual interface

```
device(config)# interface ve 10
device(config-vif-10)# ip address 10.10.10.1/24
```

f) Configure the cluster

```
device(config)# cluster bfd 10
device(config-cluster-bfd)# rbridge-id 11
device(config-cluster-bfd)# session-vlan 10
device(config-cluster-bfd)# member-vlan 7
device(config-cluster-bfd)# icl icl ethernet 1/2
device(config-cluster-bfd)# peer 10.10.10.2 rbridge-id 10 icl icl
device(config-cluster-bfd)# deploy
device(config-cluster-bfd)# client "client_1"
device(config-cluster-bfd-client-client_1)# rbridge-id 2
device(config-cluster-bfd-client-client_1)# client-interface ethernet 4/17
device(config-cluster-bfd-client-client_1)# deploy
device(config-cluster-bfd-client-client_1)# exit
device(config-cluster-bfd)# exit
```

2. Configure applications that use BFD on the device.

   Configure OSPF on global mode

```
device(config)# router ospf
device(config-ospf-router)# area 0.0.0.0
device(config-ospf-router)# bfd all-interfaces
```

3. Configure BFD parameters on the device.

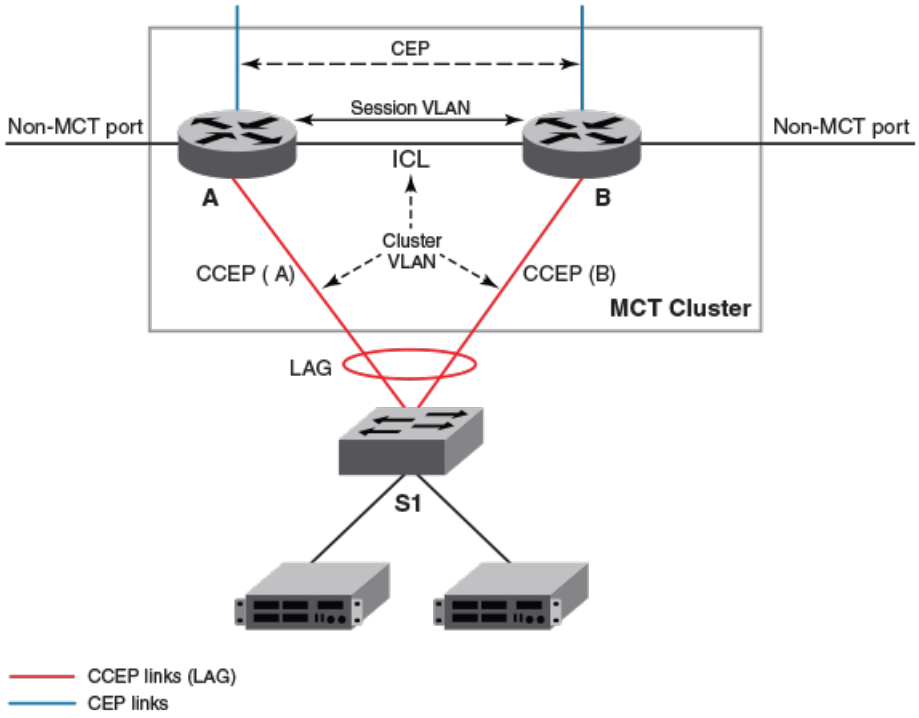
   Assign the IP address and enable BFD on the member VLAN virtual interface

```
device(config)# interface ve 7
device(config-vif-7)# ip address 10.20.20.2/24
device(config-vif-7)# bfd interval 300 min-rx 300 multiplier 3
device(config-vif-7)# ip ospf area 0.0.0.0
device(config-vif-7)# ip ospf bfd
```

BFD over MCT configuration example

The following example configures BFD over MCT on a Brocade device.

FIGURE 179 BFD over MCT



The following example illustrates the MCT configuration at router A of the network diagram.

```
lag "ccep" static id 2
 ports ethernet 1/1 ethernet 1/4
 primary-port 1/1
 deploy
!
lag "icl" static id 1
 ports ethernet 1/2 ethernet 4/5
 primary-port 1/2
 deploy
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/2 ethe 4/5
!
vlan 7 name CLIENT
 tagged ethe 1/1 to 1/2 ethe 1/4 ethe 4/5
 router-interface ve 7
 static-mac-address 0024.3843.6d41 ethernet 1/1
!
vlan 10 name ICL
 tagged ethe 1/2 ethe 4/5
 router-interface ve 10
!
no route-only
!
router ospf
 area 0.0.0.0
 bfd all-interfaces
!
interface ve 7
 bfd interval 300 min-rx 300 multiplier 3
 ip ospf area 0.0.0.0
 ip ospf bfd
 ip address 192.0.6.0/24
!
interface ve 10
 ip address 10.10.10.1/24
!
cluster "bfd" 10
 rbridge-id 11
 session-vlan 10
 cluster-client-static-mac-move
 member-vlan 7
 icl icl ethernet 1/2
 peer 10.10.10.2 rbridge-id 10 icl icl
 deploy
 client "CES-3"
  rbridge-id 2
  client-interface ethernet 1/1
  deploy
!
end
```

The following example illustrates the MCT configuration at router B of the network diagram.

```
lag "ccep" static id 2
 ports ethernet 1/1 ethernet 1/3
 primary-port 1/1
 deploy
!
lag "icl" static id 1
 ports ethernet 1/2 ethernet 3/5
 primary-port 3/5
 deploy
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/2 ethe 3/5
!
vlan 7
 tagged ethe 1/1 to 1/3 ethe 3/5
 router-interface ve 7
 static-mac-address 0024.3843.6d41 ethernet 1/1
!
vlan 10
 tagged ethe 1/2 ethe 3/5
 router-interface ve 10
!
router ospf
 area 0.0.0.0
!
interface ve 7
 bfd interval 300 min-rx 300 multiplier 3
 ip ospf area 0.0.0.0
 ip ospf bfd
 ip address 192.0.16.0/24
!
interface ve 10
 ip address 10.10.10.2/24
!
cluster "bfd" 10
 rbridge-id 10
 session-vlan 10
 cluster-client-static-mac-move
 member-vlan 7
 icl icl ethernet 3/5
 peer 10.10.10.1 rbridge-id 11 icl icl
 deploy
 client "CES-3"
  rbridge-id 2
  client-interface ethernet 1/1
  deploy
!
end
```

The following example illustrates the MCT configuration at S1 of the network diagram.

```
lag "client" static id 1
 ports ethernet 1/13 to 1/16
 primary-port 1/13
 deploy
!
!
vlan 1 name DEFAULT-VLAN
 no untagged ethe 1/5 to 1/6
!
vlan 7
 tagged ethe 1/13 to 1/16
 router-interface ve 7
!
router ospf
 area 0.0.0.0
!
interface ve 7
 bfd interval 300 min-rx 300 multiplier 3
 ip ospf area 0.0.0.0
 ip ospf bfd
 ip address 192.0.10.0/24
!
end
```

Displaying BFD information

Various show commands can be used to display BFD information.

Use one or more of the following commands to display BFD information. The commands do not have to be entered in this order.

1. Enter the **show bfd** command to display current registered protocol, BFD state and the number of BFD sessions available.

```
device(config)# show bfd
 BFD State: ENABLED Version: 1 Use PBIF Assist: Y SH setup delay 180 MH setup delay 0
 Current Registered Protocols: ospf/0
 All Sessions: Current: 2 Maximum Allowed: 250 Maximum Exceeded Count: 0
 Maximum TX/RX Sessions Allowed on LP: 80  Maximum Session Exceeded Count for LPs: 0
   LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions
   1   0/0           2   0/0           3   1/1           4   1/1
 BFD Enabled ports count: 1
 Port       MinTx       MinRx       Mult Sessions
 ve 7       300         300         3    2
```

2. Enter the **show bfd neighbor** command to display the total number of neighbor entries, neighbor address, and the current state of BFD.

```
device(config)# show bfd neighbors
Total Entries:2 R:RXRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress                 State  Interface       Holddown   Interval  R/H
20.20.20.3                      UP     ve 7            900000     300000    Y/S
20.20.20.2                      UP     ve 7            900000     300000    Y/S
```

3. Enter the **show bfd neighbor details** command to display information about the Tx and Rx ports where BFD control messages are received from the remote peer.

```
device(config)# show bfd neighbors details
Total Entries:2 R:RXRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress                 State  Interface       Holddown   Interval  R/H
20.20.20.3                      UP     ve 7            900000     300000    Y/S
  Registered Protocols(Protocol/VRFID): ospf/0
  Local: Disc: 5, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
  Remote: Disc: 1, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
```

```
  Stats: RX: 3648 TX: 4249 SessionUpCount: 1 at SysUpTime: 0:16:29:10.591
  Session Uptime: 0:0:16:36.666, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 4/17,RX: eth 4/17,Vlan Id: 7
  Using PBIF Assist: Y
NeighborAddress                       State  Interface       Holddown   Interval  R/H
20.20.20.2                            UP     ve 7            900000     300000    Y/S
  Registered Protocols(Protocol/VRFID): ospf/0
  Local: Disc: 6, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
  Remote: Disc: 6, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 300000, MinRxInterval: 300000, Multiplier: 3
  Stats: RX: 4530 TX: 4071 SessionUpCount: 1 at SysUpTime: 0:16:29:10.591
  Session Uptime: 0:0:16:36.666, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 3/3,RX: eth 3/3,Vlan Id: 7
  Using PBIF Assist: Y
```

4. Enter the **show bfd applications** command to display information about the registered protocol count.

```
device(config)# show bfd applications
  Registered Protocols Count: 1
     Protocol  VRFID     Parameter HoldoverInterval
     ospf      0         0         0
```

Configuring IP

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

Basic configuration consists of adding IP addresses and enabling a route exchange protocol. Refer to Configuring IP addresses on page 694.

To change some of the IP parameters from their default values or to view configuration information or statistics, refer to the following sections:

- The IP packet flow on page 665
- Basic IP parameters and defaults on page 669
- Configuring IP parameters on page 694

The IP packet flow

Figure 183 shows how an IP packet moves through a Brocade device.

FIGURE 180 IP Packet flow through a Brocade device

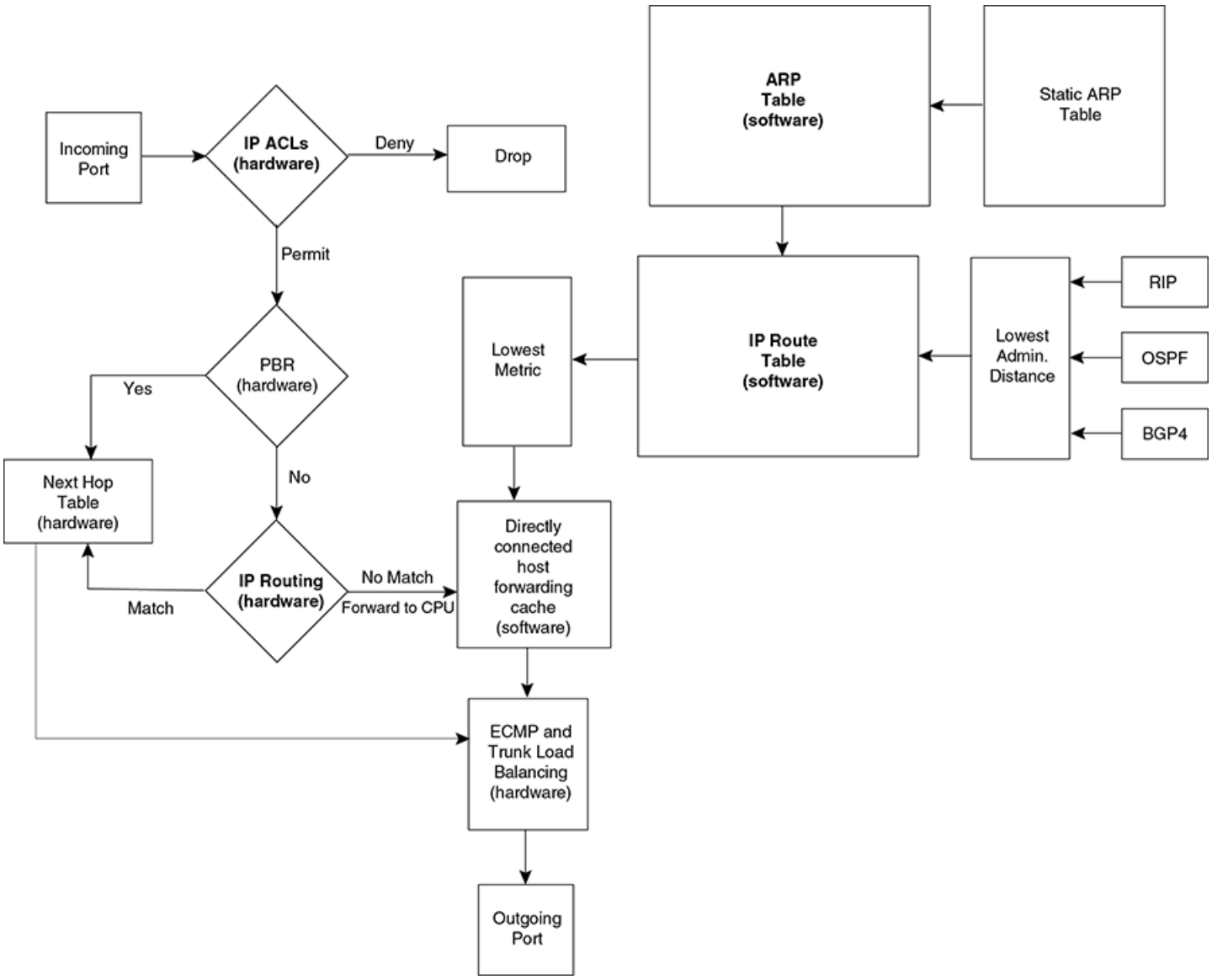


Figure 183 shows the following packet flow.

1. When the Brocade device receives an IP packet, the Brocade device checks for IP ACL filters on the receiving interface. If a deny filter on the interface denies the packet, the Brocade device discards the packet and performs no further processing. If logging is enabled for the filter, then the Brocade device generates a Syslog entry and SNMP trap message.
2. If the packet is not denied, the Brocade device checks for Policy Based Routing (PBR). If the packet matches a PBR policy applied on the incoming port, the PBR processing is performed and either drops the packet or forwards it to a port, based on the route map rules.
3. If the incoming packet does not match PBR rules, the Brocade device looks in the hardware IP routing table to perform IP routing. The hardware routing table is pre-loaded with the complete routing table, except for the directly connected host entries. Default and statically defined routes are also pre-loaded in the hardware routing table. If the incoming packet matches a route

entry, the packet is routed according to the information provided in the route entry. The ECMP and LAG load balancing is done by the hardware, if needed, to select the outgoing port.

4. If there is no match in the IP routing table and a default route is not configured, the packet is dropped. For an IP packet whose destination IP address is to a directly connected host, the first packet is forwarded to the CPU. If the ARP is resolved and the host is reachable, the CPU creates a route entry in the hardware to route subsequent packets in hardware.

   The software enables you to display the ARP cache and static ARP table, the IP route table, the IP forwarding cache.

   You also can change the capacity of the following tables by changing the memory allocation for the table:

   - ARP cache table on page 667
   - Static ARP table on page 667
   - IP route table on page 668
   - IP forwarding cache on page 668

ARP cache table

The Address Resolution Protocol (ARP) is supported on the Brocade device. Refer to Configuring ARP parameters on page 713.

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

An exception is an ARP entry for an interface-based static IP route that goes to a destination that is one or more router hops away. For this type of entry, the MAC address is either the destination device's MAC address or the MAC address of the router interface that answered an ARP request on behalf of the device, using proxy ARP.

The ARP cache can contain dynamic (learned) entries and static (user-configured) entries. The software places a dynamic entry in the ARP cache when the Brocade device learns a device's MAC address from an ARP request or ARP reply from the device.

The software can learn an entry when the Brocade device receives an ARP request from another IP forwarding device or an ARP reply.

: Dynamic entry

```
        IP Address          MAC Address         Type        Age     Port
1       10.95.6.102         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 Brocade device has a static ARP table.

Entries in the static ARP table are user-configured. You can add entries to the static ARP table regardless of whether the device the entry is for is connected to the Brocade device.

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

: Static ARP entry

```
    Index   IP Address          MAC Address         Port
    1       10.95.6.111         0800.093b.d210      1/1
```

Each entry lists the information you specified when you created the entry.

To display ARP entries, refer to the following:

- Displaying the ARP cache on page 793
- Displaying the static ARP table on page 794

To configure other ARP parameters, refer to Configuring ARP parameters on page 713.

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

- For dynamic entries, refer to the "Displaying and modifying default settings for system parameters". The ip-arp parameter controls the ARP cache size.

IP route table

The IP route table contains paths to IP destinations.

The IP route table can receive the paths from the following sources:

- A directly-connected destination, which means there are no router hops to the destination
- A static IP route, which is a user-configured route
- A route learned through RIP
- A route learned through OSPF
- A route learned through IS-IS
- 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.

: IP route table

```
Destination         Gateway           Port     Cost    Type    Uptime
10.0.0.0/8          10.20.176.1       mgmt 1    1/1     S      11m59s
```

Each IP route table entry contains the destination's IP address and subnet mask and the IP address of the next-hop router interface to the destination. Each entry also indicates the port attached to the destination or the next-hop to the destination, the route's IP metric (cost), and the type. The type indicates how the IP route table received the route.

To display the IP route table, refer to Displaying the IP route table on page 796.

To configure a static IP route, refer to Configuring static routes on page 745.

To clear a route from the IP route table, refer to Clearing IP routes on page 800.

To increase the size of the IP route table for learned and static routes, refer to "Displaying and modifying default settings for system parameters".

.Consider the following:

- For learned routes, modify the ip-route parameter.
- For static routes, modify the ip-static-route parameter.

IP forwarding cache

The Brocade device maintains a software cache table for fast processing of IP packets that are forwarded or generated by the CPU. The cache also contains forwarding information that is normally contained in the IP routing table. For example, the cache contains information on the physical outgoing port, priority, VLAN, and the type of cache entry. Also, cache entries have hardware information, which is useful for debugging and aging.

There are two types of IP cache entries.

1. Directly connected host entries - These entries are created when the CPU receives the first packet destined to a directly connected host. Host entries are set to age out after a certain period if no traffic is seen for that entry.
2. Network entries - These entries are created when a route table entry is created in software. These entries are not subjected to aging. A route table entry is created when routes are learned by routing protocols such as OSPF or when routes are statically configured.

   : 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 Brocade device itself, as shown here, then next-hop information indicates this. The port through which the destination is reached is also listed, as well as the VLAN and Layer 4 QoS priority associated with the destination if applicable.

   To display the IP forwarding cache, refer to Displaying the forwarding cache on page 795.

IP packet queuing

When the user wants to send a packet to a local host, the software looks up the IP in the ARP cache. If the address is found, it gets the MAC address, constructs an Ethernet header with the correct source or destination MAC addresses, and sends it.

If the address is not found in the table, ARP broadcasts a packet to every host on the Ethernet, except the one from which it received the packet. The packet contains the IP address for which an Ethernet address is sought. If a receiving host identifies the IP address as its own, it will send its Ethernet address back to the requesting host.

For management of IP packet queuing when a packet is received for a directly connected host when there is no MAC address available, the **ip drop-arp-pending-packets** command has been added to allow the packets in the CPU to be dropped.

To set all packets in the LP buffer to be dropped when ARP resolution is going on, enter a command such as the following:

```
device(confi
g)#ip drop-arp-pending-packets
```

Syntax: [no] ip drop-arp-pending-packets

Use the**no ip drop-arp-pending-packets command** to return to the default behavior of continue with pending IP packets while ARP resolution.

Basic IP parameters and defaults

IP is enabled by default. The following protocols are disabled by default:

- Route exchange protocols (RIP, OSPF, IS-IS, BGP4)
- Multicast protocols (IGMP, PIM-DM, PIM-SM)
- Router redundancy protocols (VRRP-E, 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 79 lists the IP global parameters for the Brocade device, their default values, and where to find configuration information.

TABLE 79 IP global parameters

Parameter	Description	Default
IP state	The Internet Protocol, version 4	Enabled **NOTE** You cannot disable IP.
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.
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.
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
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
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
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.	Ten minutes

TABLE 79 IP global parameters (continued)

Parameter	Description	Default
	NOTE You also can change the ARP age on an individual interface basis. Refer to IP interface parameters on page 673.	
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
Static ARP entries	An ARP entry you place in the static ARP table. Static entries do not age out.	No entries
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
Directed broadcast forwarding	A directed broadcast is a packet containing all ones (or in some cases, all zeros) in the host portion of the destination IP address. When a router forwards such a broadcast, it sends a copy of the packet out each of its enabled IP interfaces. **NOTE** You also can enable or disable this parameter on an individual interface basis. Refer to IP interface parameters on page 673.	Disabled
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.
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
Internet Control Message Protocol (ICMP) messages	The Brocade device can send the following types of ICMP messages: • Echo messages (ping messages) • Destination Unreachable messages • Redirect messages **NOTE** You also can enable or disable ICMP Redirect messages on an individual interface basis. Refer to IP interface parameters on page 673.	Enabled
ICMP Router Discovery Protocol (IRDP)	An IP protocol a router can use to advertise the IP addresses of its router interfaces to directly	Disabled

TABLE 79 IP global parameters (continued)

Parameter	Description	Default
	attached hosts. You can enable or disable the protocol, and change the following protocol parameters: • Forwarding method (broadcast or multicast) • Hold time • Maximum advertisement interval • Minimum advertisement interval • Router preference level NOTE You also can enable or disable IRDP and configure the parameters on an individual interface basis. Refer to IP interface parameters on page 673.	
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
Maximum Frame Size	You can set a maximum frame size of all Ethernet frames that are forwarded by the system.	
Domain name for Domain Name Server (DNS) resolver	A domain name (example: brocade.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 device.	None configured
DNS default gateway addresses	A list of gateways attached to the device through which clients attached to the device can reach DNS.	None configured
IP load sharing	A feature that enables the device 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
Maximum IP load sharing paths	The maximum number of equal-cost paths across which the Brocade device is allowed to distribute traffic.	Four
Origination of default routes	You can enable a device to originate default routes for the following route exchange protocols, on an individual protocol basis: • RIP • OSPF • BGP4	Disabled
Default network route	The device 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

TABLE 79 IP global parameters (continued)

Parameter	Description	Default
Static route	An IP route you place in the IP route table.	No entries
Source interface	The IP address the device uses as the source address for Telnet, RADIUS, or TACACS/TACACS+ packets originated by the device. The device 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.

IP interface parameters

Table 80 lists the interface-level IP parameters for the Brocade device, their default values, and where to find configuration information.

TABLE 80 IP interface parameters

Parameter	Description	Default
IP state	The Internet Protocol, version 4	Enabled **NOTE** You cannot disable IP.
IP address	A Layer 3 network interface address The Brocade device has separate IP addresses on individual interfaces.	None configured
Encapsulation type	The format of the packets in which the device encapsulates IP datagrams. The encapsulation format can be one of the following: • Ethernet II • SNAP	Ethernet II
IP Maximum Transmission Unit (MTU)	The maximum length (number of bytes) of an encapsulated IP datagram the device can forward.	1500 for Ethernet II encapsulated packets 1492 for SNAP encapsulated packets
ARP age	Locally overrides the global setting. Refer to IP global parameters on page 670.	Ten minutes
Directed broadcast forwarding	Locally overrides the global setting. Refer to IP global parameters on page 670.	Disabled
ICMP Router Discovery Protocol (IRDP)	Locally overrides the global IRDP settings. Refer to IP global parameters on page 670.	Disabled
ICMP Redirect messages	Locally overrides the global setting. Refer to IP global parameters on page 670.	Enabled
DHCP gateway stamp	The device 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 device interface that receives the request in the request packet's Gateway field.	The lowest-numbered IP address on the interface that receives the request

TABLE 80 IP interface parameters (continued)

Parameter	Description	Default
	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.	
UDP broadcast forwarding	The device can forward UDP broadcast packets for UDP applications such as BootP. By forwarding the UDP broadcasts, the device enables clients on one subnet to find servers attached to other subnets. **NOTE** To completely enable a client's UDP application request to find a server on another subnet, you must configure an IP helper address consisting of the server's IP address or the directed broadcast address for the subnet that contains the server. Refer to the next row.	The device helps forward broadcasts for the following UDP application protocols: • bootps • dns • netbios-dgm • netbios-ns • tacacs • tftp • time
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 device to forward requests for certain UDP applications from a client on one subnet to a server on another subnet.	None configured

GRE IP tunnel

Multi-Service IronWare software supports the tunneling of packets with the Generic Routing Encapsulation (GRE) mechanism over an IP network, as described in RFC 2784. With GRE, packets are encapsulated in a transport protocol packet at a tunnel source and delivered to a tunnel destination, where they are unpacked and made available for delivery.

Considerations in implementing this feature

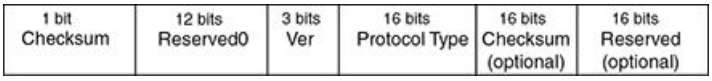
The considerations in implementing this feature are as follows:

- As a point-to-point tunnel configuration, GRE requires both ends of the tunnel to be configured.
- Only four-byte GRE headers are supported at the ingress (even though eight-byte headers can be processed at a transit node or the egress point).
- A Brocade device does not support the key and sequence numbering option with GRE (per RFC 2890).
- The current maximum number of tunnels is 8192 (with default as 256 tunnels).

  NOTE
  Do not forward packets from one type of tunnel to another type of tunnel in XPP. Packets may not be routed properly.

Figure 184 describes the GRE header format.

FIGURE 181 GRE header format



Checksum - This field is assumed to be zero in this version. If set to 1 means that the **Checksum** (optional) and **Reserved** (optional) fields are present and the Checksum (optional) field contains valid information.

Reserved0 - Bits 6:0 of the field are reserved for future use and must be set to 0 in transmitted packets. If bits 11:7 of the field are non-0, then a receiver must discard the packet. This field is assumed to be 0 in this version.

Ver - This field must be set to 0. This field is assumed to be 0 in this version.

Protocol Type - This field contains the EtherType of the payload protocol.

For details on configuring a GRE IP tunnel, refer Examples on page 781.

GRE MTU enhancements

Enhancements have been introduced to support GRE MTU in support of RFC 4459. This includes support for the following:

- Signaling the Lower MTU to the Sources as described in Section 3.2 of RFC 4459
- Fragmentation of the Inner packet as described in Section 3.4 of RFC 4459

This enhancement also allows you to set a specific MTU value for packets entering a configured GRE tunnel. Packets whose size is greater than the configured value are fragmented and encapsulated with IP/GRE headers for transit through the tunnel. This feature supports Jumbo packets although they may be fragmented based on the MTU value configured.

Configuring a GRE IP Tunnel

To configure a GRE IP Tunnel, configure the following parameters:

- CAM restrictions on page 676
- Maximum Number of Tunnels (optional)
- Tunnel Interface
- Source Address or Source Interface for the Tunnel
- Destination address for the Tunnel
- GRE Encapsulation
- IP address for the Tunnel
- Keep Alive Support (optional)
- TTL Value (optional)
- TOS Value (optional)
- MTU Value (optional)

Configuration considerations

1. To enable keepalive when a GRE source and destination are directly connected, you must disable ICMP redirect on the tunnel source port on the GRE nodes. Otherwise, the keepalive packets go to the CPU where they can degrade CPU performance.
2. Whenever multiple IP addresses are configured on a tunnel source, the primary address of the tunnel is always used for forming the tunnel connections. Consequently, you must carefully check the configurations when configuring the tunnel destination.
3. GRE tunneling is not supported for non-default VRFs.
4. When a GRE tunnel is configured, you cannot configure the same routing protocol on the tunnel through which you learn the route to the tunnel destination. For example, if the Brocade device learns the tunnel destination route through OSPF protocol, you cannot configure the OSPF protocol on the same Tunnel and vice-versa. When a tunnel has OSPF configured, the Brocade device cannot learn the tunnel destination route through OSPF. This could cause the system to become unstable.

NOTE
With GRE Dynamic-cam mode, at the Egress node, when a GRE packet is received, the Brocade device programs the CAM entries to forward the packets based on Inner DPA. These host CAM entries will be aging even if the traffic is hitting that CAM entries. This will cause the CAM entries to become aged out and recreated which could cause a small packet loss.

Configuring ECMP for routes through an IP GRE tunnel

If multiple routes are using IP GRE tunnels to a destination, packets are automatically load-balanced between tunnels. This feature allows for load distribution of traffic among the available IP GRE tunnels. If the routes to a destination are both normal IP routes and routes through IP GRE tunnels, ECMP is not enabled.

CAM restrictions

CAMs are partitioned on a Brocade device by a variety of profiles that you can select for your specific application.

To implement a CAM partition for a GRE tunnel, enter a command such as the following.

```
device(config)# cam-partition profile ipv4
```

Syntax: [no] cam-partition profile { **ipv4** | **ipv4-ipv6** | **ipv4-vpls** | **ipv4-vpn** | **ipv6** | **l2-metro** | **l2-metro-2** | **mpls-l3vpn** | **mpls-l3vpn-2** | **mpls-vpls** | **mpls-vpls-2** | **mpls-vpn-vpls** | **multi-service** | **multi-service-2** | **multi-service-3** | **multi-service-4** }

The **ipv4** parameter adjusts the CAM partitions, as described in the tables below, to optimize the device for IPv4 applications.

NOTE
The **ipv4** parameter is effective only if you first entered the following command:

```
device(config)# system-max ipv6-mcast-cam 0
```

The **ipv4-ipv6** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for IPv4 and IPv6 dual stack applications.

The **ipv4-vpls** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for IPv4 and MPLS VPLS applications.

The **ipv4-vpn** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for IPv4 and MPLS Layer-3 VPN applications.

The **ipv6** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for IPv6 applications.

The **l2-metro** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for Layer 2 Metro applications.

The **l2-metro-2** parameter provides another alternative to **l2-metro** to optimize the device for Layer 2 Metro applications. It adjusts the CAM partitions, as described in the tables below for the Brocade NetIron XMR Series.

The **mpls-l3vpn** parameter adjusts the CAM partitions, as described in the tables below, to optimize the device for Layer 3, BGP or MPLS VPN applications.

The **mpls-l3vpn-2** parameter provides another alternative to **mpls-l3vpn** to optimize the device for Layer 3, BGP or MPLS VPN applications.

The **mpls-vpls** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for MPLS VPLS applications.

The **mpls-vpls-2** parameter provides another alternative to **mpls-vpls** to optimize the device for MPLS VPLS applications. It adjusts the CAM partitions, as described in the tables below.

The **mpls-vpn-vpls** parameter adjusts the CAM partitions, as described in the tables below, to optimize the device for MPLS Layer-3 and Layer-2 VPN applications.

The **multi-service** parameter adjusts the CAM partitions, as described in the tables below to optimize the device for Multi-Service applications.

The **multi-service-2** parameter provides another alternative to multi-service to optimize the device for Multi-Service applications.

The **multi-service-3** parameter provides another alternative to multi-service to optimize the device for Multi-Service applications to support IPv6 VRF.

The **multi-service-4** parameter provides another alternative to multi-service to optimize the device for Multi-Service applications to support IPv6 VRF.

CAM partition profiles for the Brocade NetIron XMR Series and Brocade NetIron MLX Series

Not all CAM profiles are compatible for running Layer 2 switching and configuring GRE tunnels simultaneously on a Brocade device.

TABLE 81 Partition profiles for the Brocade NetIron XMR Series and Brocade NetIron MLX Series.

Compatible CAM profiles
default
ipv4
ipv6
ipv4-vpn
mpls-l3vpn
mpls-l3vpn-2
multi-service-2 (Does not support GRE tunnel with VE interface as the source address in this profile.)
multi-service-3 (Does not support GRE tunnel with VE interface as the source address in this profile.)
multi-service-4 (Does not support GRE tunnel with VE interface as the source address in this profile.)

Configuring the maximum number of tunnels supported

You can configure the devices to support a specified number of tunnels using the following command.

```
device(config)# system-max ip-tunnels 512
device(config)# write memory
```

Syntax: system-max ip-tunnels number

The *number* variable specifies the number of GRE tunnels that can be supported.

The Brocade NetIron XMR Series and Brocade NetIron MLX Series permissible range is 1 - 8192. The default value is 256. The permissible range for Brocade NetIron CES Series devices is 32 - 128. The default value is 32. The permissible range for Brocade NetIron CER Series devices is 32 - 256. The default value is 32.

NOTE
Multicast over GRE tunnels for PIM can support up to the default system max of 256 tunnels if the required hardware resources are available.

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

Configuring a tunnel interface

To configure a tunnel interface, use a the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)
```

Syntax: [no] interface tunnel tunnel id

The *tunnel-id* variable is numerical value that identifies the tunnel being configured.Possible range is from 1 to the maximum configured tunnels in the system.

Configuring a source address or source interface for a tunnel interface

To configure a source address for a specific tunnel interface, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel source 10.0.8.108
```

To configure a source interface for a specific tunnel interface, enter the following command.

```
device(config)# interface tunnel 100
device(config-tnif-100)tunnel source ethernet 3/1
```

Syntax: [no] tunnel source ip-address | port-no

You can specify either of the following:

The *ip-address* variable is the source IP address being configured for the specified tunnel.The*port-no* variable is the source slot or port of the interface being configured for the specified tunnel. When you configure a source interface, there must be at least one IP address configured on that interface. Otherwise, the interface will not be added to the tunnel configuration and an error message like the following will be displayed: Error - Tunnel source interface 3/1 has no configured ip address.

It can be a physical or virtual interface (ve).

Configuring a destination address for a tunnel interface

To configure a destination address for a specific tunnel interface, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel destination 10.108.5.2
```

Syntax: [no] tunnel destination ip-address

The *ip-address* variable is destination IP address being configured for the specified tunnel.

NOTE
If GRE is configured with a tunnel destination reachable over LAG ports, load balancing will only work with the following LAG types: server LAG or LACP with server LAG. Traffic cannot be load-balanced across multiple ports of a switch LAG.

NOTE
Traffic from a GRE tunnel entering a MPLS tunnel is not supported.

Configuring a tunnel interface for GRE encapsulation

To configure a specified tunnel interface for GRE encapsulation, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel mode gre ip
```

Syntax: [no] tunnel mode gre ip

The **gre** parameter specifies that the tunnel will use GRE encapsulation

The **ip** parameter specifies that the tunnel protocol is IP.

Configuring an IP address for a tunnel interface

To configure an IP address for a specified tunnel interface, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)ip address 10.10.3.1/24
```

Syntax: [no] ip address ip-address

The *ip-address* variable is the IP address being configured for the specified tunnel interface.

Configuring keep alive support

This parameter is optional. It lets the device maintain a tunnel in an up or down state based upon the periodic sending of keep alive packets and the monitoring of responses to the packet. If the packets fail to reach the tunnel's far end more frequently than the configured number of retries, the tunnel is placed in a down state. A keep alive packet is a GRE IP packet with no payload.

To configure the keep alive option, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)keepalive 5 4
```

Syntax: [no] keepalive seconds retries

The *seconds* variable specifies the number of seconds between each initiation of a keep alive message. The range for this interval is 1 - 32767 seconds. The default value is 10 seconds.

The *retries* variable specifies the number of times that a packet is sent before the system places the tunnel in the down state. Possible values are from 1 - 255. The default number of retries is 3.

Configuring a TTL value

This is an optional parameter that allows you to set the Time-to-Live value for the outer IP header of the GRE tunnel packets.

To configure the TTL value, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel ttl 100
```

Syntax: [no] tunnel ttl ttl-value

The *ttl--value* variable specifies a TTL value for the outer IP header. Possible values are 1 - 255. The default value is 255.

Configuring a TOS value

This is an optional parameter that allows you to set the TOS value for the outer IP header of the GRE tunnel packets.

To configure the TOS value, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel tos 100
```

Syntax: [no] tunnel tos tos-value

The *tos-value* variable specifies a TOS value for the outer IP header.

The Brocade NetIron XMR Series and Brocade NetIron MLX Series possible values are 0 - 255. The default value is 0.

The Brocade NetIron CES Series and Brocade NetIron CER Series devices possible values are 0 - 63. The default value is 0.

Configuring GRE session enforce check

The **gre-session-enforce-check** command lets you enable the GRE session enforce check. When a GRE packet arrives and this feature is enabled, the system tries to match the GRE packet source and destination address pair with the tunnel configured destination and source pair. If the pairs do not match, the packet is dropped in the hardware. The default behavior when this command is disabled is to terminate the GRE tunnel based on the destination IP address.

> NOTE
> The Brocade NetIron CES Series and Brocade NetIron CER Series devices currently do not support the **ip-tunnel-policy** and the **accounting-enable commands.**

To configure the GRE session enforce check, go to the IP tunnel policy context, and then enter the **gre-session-enforce-check** command.

```
device(config)#ip-tunnel-policy
device(config-ip-tunnel-policy)#gre-session-enforce-check
```

Syntax: [no] gre-session-enforce-check

To disable the GRE session enforce check, use the **no** form of this command. This command is disabled by default. You might have to write the configuration to memory and reload the system whenever the configuration of this command is changed because a one-time creation of a source-ingress CAM partition is necessary. The system prompts you if the memory write and reload are required.

The first-time execution of certain commands necessitates the creation of a source-ingress CAM partition, after which you write to memory and reload. These commands are **gre-session-enforce-check** , **ipv6-session-enforce-check** , and **accounting-enable** . After this CAM partition is created, it is not necessary to follow either of the other two commands with a memory write and reload. The CAM partition is created out of the Layer 4 CAM and has no impact on the Layer 3 route scalability.

Configuring a maximum MTU value for a tunnel interface

You can set an MTU value for packets entering the tunnel. Packets that exceed either the default MTU value of 1476 bytes or the value that you set using this command are fragmented for transit through the tunnel. The default MTU value is set to 1476.

> NOTE
> The tunnel MTU should be configured explicitly for packet size greater than 1476 bytes.

The following command allows you to change the MTU value for packets transiting "tunnel 1".

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel mtu 1500
```

Syntax: [no] tunnel mtu packet-size

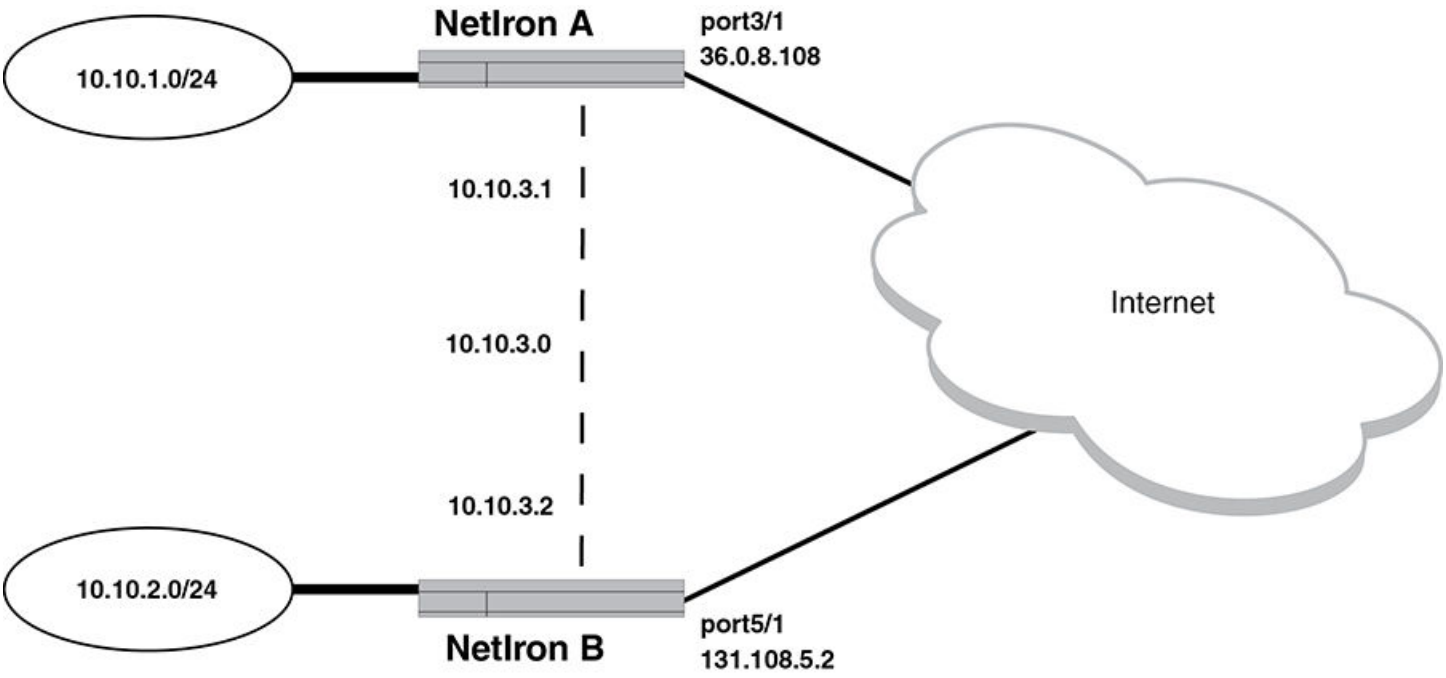
The *packet-size* variable specifies the maximum MTU size in bytes for the packets transiting the tunnel.

> **NOTE**
> To prevent packet loss after the 24 byte GRE header is added, make sure that any physical interface that is carrying GRE tunnel traffic has an IP MTU setting at least 24 bytes greater than the tunnel MTU setting.

Example of a GRE IP tunnel configuration

In this example, a GRE IP Tunnel is configured between the Brocade A device and the Brocade B device. Traffic between networks 10.10.1.0/24 and 10.10.2.0/24 is encapsulated in a GRE IP packet sent through the tunnel on the 10.10.3.0 network. and unpacked and sent the destination network. A static route is configured at each device to go through the tunnel interface to the target network.

FIGURE 182 GRE IP tunnel configuration example



Configuration example for Brocade A

```
device(config)# interface ethernet 3/1
device(config-int-e10000-3/1)# ip address 36.0.8.108/24
device(config)# interface tunnel 1
device(config)# vrf forwarding red
device(config-tnif-1)# tunnel source 36.0.8.108
device(config-tnif-1)# tunnel destination 131.108.5.2
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# ip address 10.10.3.1/24
device(config-tnif-1)# int loopback 1
device(config-tnif-1)# vrf forwarding red
device(config-tnif-1)# ip address 10.10.1.1/32
```

```
device(config-tnif-1)# keepalive 5 4
device(config-tnif-1)# vrf red
device(config-tnif-1)# rd 1:1
device(config-tnif-1)# address-family ipv4
device(config-tnif-1)# ip route 10.10.2.0/24 10.10.3.2
device(config-tnif-1)# exit
device(config)# ip route 10.10.2.0/24 10.10.3.2
```

Configuration example for Brocade B

```
device(config)# interface ethernet 5/1
device(config-if-e10000-5/1)# ip address 131.108.5.2/24
device(config)# interface tunnel 1
device(config)# vrf forwarding red
device(config-tnif-1)# tunnel source ethernet 5/1
device(config-tnif-1)# tunnel destination 36.0.8.108
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# ip address 10.10.3.2/24
device(config-tnif-1)# int loopback 1
device(config-tnif-1)# vrf forwarding red
device(config-tnif-1)# ip address 10.10.1.1/32
device(config-tnif-1)# keepalive 5 4
device(config-tnif-1)# vrf red
device(config-tnif-1)# rd 2:2
device(config-tnif-1)# address-family ipv4
device(config-tnif-1)# ip route 10.10.2.0/24 10.10.3.2
device(config-tnif-1)# exit
device(config)# ip route 10.10.2.0/24 10.10.3.2
```

NOTE

Traffic from a GRE tunnel entering a MPLS tunnel is not supported.

Displaying GRE tunneling information

You can display GRE tunneling information using the **show ip interface**,**show ip route** and **show interface tunnel** commands as shown in the following.

```
device# show ip interface tunnel 1
Interface Tunnel 1
  port enabled
  port state: UP
  ip address: 10.255.255.13/24
  Port belongs to VRF: red
  encapsulation: ETHERNET, mtu: 1476
  directed-broadcast-forwarding: disabled
  ip icmp redirect: enabled
  No inbound ip access-list is set
  No outbound ip access-list is set
  No Helper Addresses are configured.
```

Syntax: show ip interface tunnel tunnel-no

The **show ip route** command displays routes that are pointing to a GRE tunnel as shown in the following.

Syntax: show interface tunnel tunnel-no

```
device# show ip route
     Total number of IP routes: 2
     Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost -Dist/Metric
             Destination        Gateway              Port          Cost    Type Uptime src-vrf
1             10.10.2.0/24       10.10.3.2          gre_tnl 1        1/1           S    7h55m      -
2             10.10.3.0/24       DIRECT             gre_tnl 1        0/0           D    7h55m      -

device# show interface tunnel 1
    Tunnel1 is up, line protocol is up
    Hardware is Tunnel
    Tunnel source 10.45.3.3
    Tunnel destination is 10.45.48.1
```

```
Tunnel mode gre ip
 No port name
 Internet address is 10.255.255.13/24,
Tunnel TOS 0, Tunnel TTL 255 MTU 1476 bytes
Keepalive is not Enabled
VRF Forwarding: Red
```

Syntax: show ip tunnel tunnel-no

```
device# show ip-tunnels 1
IPv4 tnnl 1 UP : src_ip 36.0.8.108, dst_ip 131.108.5.2
TTL 255, TOS 0, NHT 0, MTU 1480, vrf: red
```

GRE tunnel VRF support

GRE tunnel VRF support maintains end - end VRF autonomy with the GRE tunnel. You can also create separate GRE tunnels on a per-VRF basis.

GRE tunnel VRF support overview

VRFs are used to segment the traffic associated with various customers of interest (CIs). These CIs are spread across geographical areas. Hence CIs enable use of GRE tunnels for non-default VRFs.

Configuration considerations

- The **vrf forwarding** command is optional. If this command is not specified, then the VRF is assumed as default VRF.
- The configured VRF must exist in the MLX device.
- Configured VRFs should be same on both nodes of the GRE tunnel, for proper working of GRE.
- Configuration is allowed for two tunnels when the tunnel destination addresses are the same and the corresponding source addresses are different. Also, configuration is allowed for two tunnels when the tunnel source addresses are the same and the corresponding destination addresses are different.
- The **vrf forwarding** configuration is supported only for GRE tunnel.
- L3VPN ID information with respect to each tunnel is configured by the **vrf forwarding** command under the tunnel interface.

Configuring the GRE VRF tunnel

Syntax: vrf forwarding *vrf-name*

Error messages

The following messages are displayed for different VRF configurations.

1. If a tunnel is configured with the VRF configuration and tunnel mode is non-GRE IP, then the following error message is displayed.
   - Error: Tunnel mode should be GRE IP/ IPSec when VRF forwarding is configured on tunnel.
2. If the tunnel source interface is on a non-supported card, then the configuration will be rejected, if the tunnel source is a physical interface or a virtual interface.
   - Error: Tunnel source interface eth 1/2 or ve103 cannot be a BR-MLX-10Gx24-DM/Gen1.1 port.
3. If the tunnel source is a loopback interface, a warning will be displayed if a BR-MLX-10Gx24-DM/Gen1.1 card is present in the chassis.
   - Warning: Tunnel source configured as loopback could be using a BR-MLX-10Gx24-DM/Gen1.1 port.
4. The VRF forwarding configuration is supported only if tunnel source is pre-configured. Otherwise, an error message is displayed.
   - Error: Please configure tunnel source before configuring tunnel VRF.

5. The VRF forwarding configuration is rejected if GRE is configured as MPLS interface and GRE is part of the VRF.
    - Error: GRE configured as MPLS interface with VRF not supported .

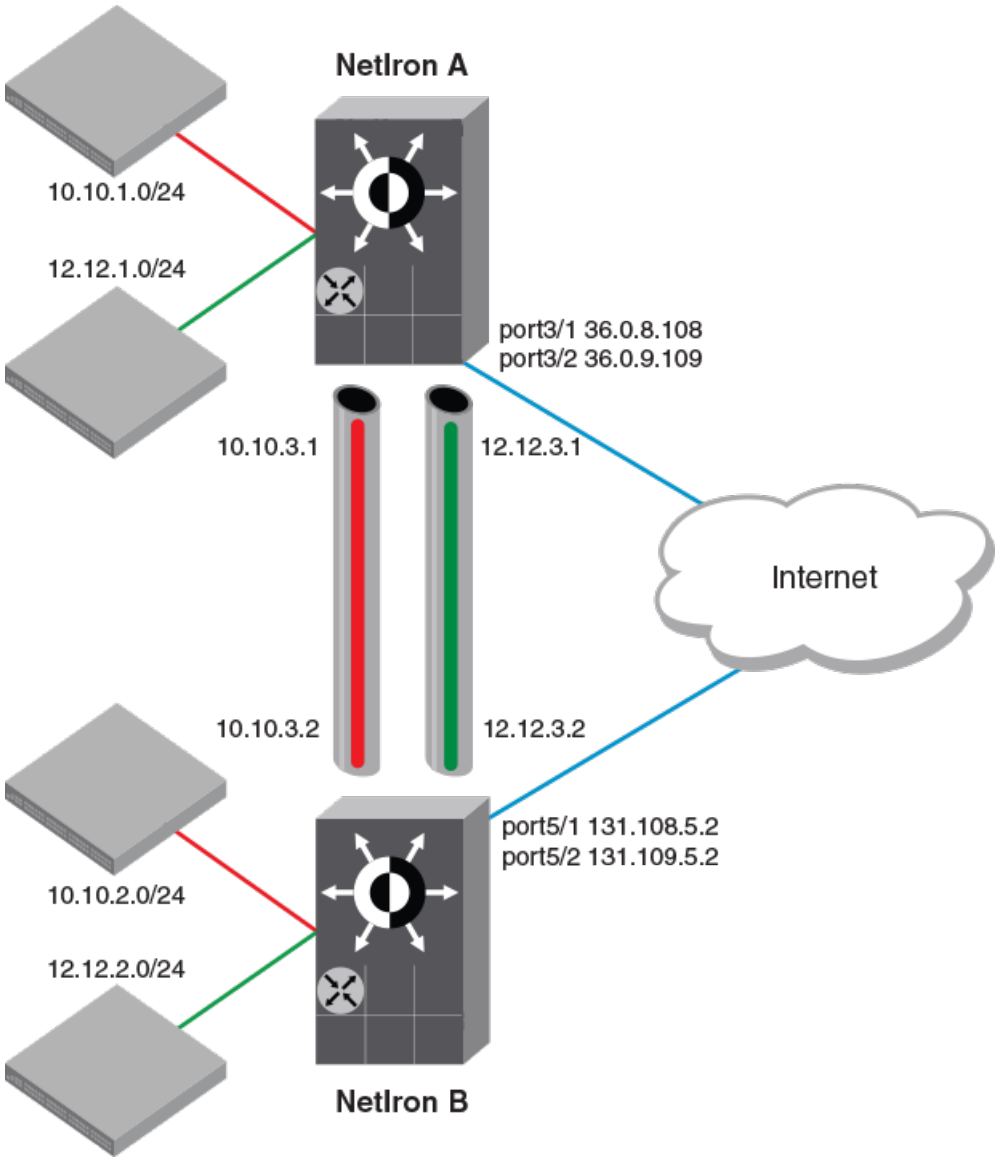
Example of a GRE VRF tunnel configuration:

In the following example, a GRE VRF tunnel is configured between the Brocade A device and the Brocade B device. Traffic between networks 10.10.1.0/24 (VRF red) and 10.10.2.0/24 (VRF red) is encapsulated in a GRE IP packet (Tunnel 1 corresponding to VRF red) sent through the tunnel on the 10.10.3.0 network and unpacked and sent to the destination network. A static route is configured at each device to go through the tunnel interface to the target network.

On the Brocade B device, the GRE tunneled packet is received in default VRF. It is unpacked and sent to the destination network on VRF red.

In this example, VRF is configured to the tunnel interface configuration using the **vrf forwarding** command (as done for all other interfaces like physical interface, the loopback interface, and so on.

GRE VRF tunnel configuration example



Configuration example for Brocade A

```
(NetIron A)

device(config)# interface eth 3/1
device(config-int-e10000-3/1)# ip address 36.0.8.108/32
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# tunnel source 36.0.8.108
device(config-tnif-1)# tunnel destination 131.108.5.2
device(config-tnif-1)# vrf forwarding  red
device(config-tnif-1)# ip address 10.10.3.1/24
device(config-tnif-1)# int loopback 1
device(config-lbif-1)# vrf forwarding red
device(config-lbif-1)# ip address 10.10.1.1/32
device(config-tnif-1)# vrf red
device(config-vrf-red)# rd 1:1
device(config-vrf-red)# address-family ipv4
```

```
device(config-vrf-red-ipv4)# ip route 10.10.2.0/24 10.10.3.2
device(config-vrf-red-ipv4)# exit-vrf

(NetIron A)

device(config)# interface eth 3/1
device(config-int-e10000-3/1)# ip address 36.0.9.108/32
device(config-tnif-1)# interface tunnel 1
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# tunnel source 36.0.9.108
device(config-tnif-1)# tunnel destination 131.109.5.2
device(config-tnif-1)# vrf forwarding  green
device(config-tnif-1)# ip address 12.12.3.1/24
device(config-tnif-1)# interface eth 3/1
device(config-if-e10000-3/1)# ip address 36.0.9.108/32
device(config-if-e10000-3/1)# int loopback 2
device(config-lbif-2)# vrf forwarding green
device(config-lbif-2)# ip address 12.12.1.1/32
device(config-tnif-1)# vrf red
device(config-vrf-red)# rd 1:1
device(config-vrf-red)# address-family ipv4
device(config-vrf-red-ipv4)# ip route 12.12.2.0/24 12.12.3.2
device(config-vrf-red-ipv4)# exit-vrf
```

Configuration example for Brocade B

```
(NetIron B)

device(config)# interface eth 3/1
device(config-int-e10000-3/1)# ip address 131.108.5.2/32
device(config)# interface tunnel 1
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# tunnel source 131.108.5.2
device(config-tnif-1)# tunnel destination 36.0.8.108
device(config-tnif-1)# vrf forwarding  red
device(config-tnif-1)# ip address 10.10.3.2/24
device(config-tnif-1)# int loopback 1
device(config-lbif-1)# vrf forwarding red
device(config-lbif-1)# ip address 10.10.2.1/32
device(config-tnif-1)# vrf red
device(config-vrf-red)# rd 2:2
device(config-vrf-red)# address-family ipv4
device(config-vrf-red-ipv4)# ip route 10.10.1.0/24 10.10.3.1
device(config-vrf-red-ipv4)# exit-vrf

(NetIron B)

device(config)# interface eth 3/1
device(config-int-e10000-3/1)# ip address 131.109.5.2/32
device(config-tnif-1)# interface tunnel 1
device(config-tnif-1)# tunnel mode gre ip
device(config-tnif-1)# tunnel source 131.109.5.2
device(config-tnif-1)# tunnel destination 36.0.9.108
device(config-tnif-1)# vrf forwarding  green
device(config-tnif-1)# ip address 12.12.3.2/24
device(config-tnif-1)# interface eth 3/1
device(config-if-e10000-3/1)# ip address 36.0.9.108/32
device(config-if-e10000-3/1)# int loopback 2
device(config-lbif-2)# vrf forwarding green
device(config-lbif-2)# ip address 12.12.2.1/32
device(config-tnif-1)# vrf red
device(config-vrf-red)# rd 2:2
device(config-vrf-red)# address-family ipv4
device(config-vrf-red-ipv4)# ip route 12.12.1.0/24 12.12.3.1
device(config-vrf-red-ipv4)# exit-vrf
```

Once the configuration is completed, the tunnel interface will come up operationally and become part of the corresponding VRF. Both MP and LP will have VRF information corresponding to the tunnel.

The route entry in that VRF shows the tunnel interface as a directly connected interface. Once a static route is configured with a destination as the CI in a VRF, the next hop will point to the corresponding tunnel interface for that VRF.

MP CPU forwarding

When the MP has to send a packet over the GRE tunnel, it creates the GRE encapsulated IP packet and sends it to the LP for transmission out of the port. The MP also supports fragmentation of packets going out of GRE.

With respect to GRE support for VRF, the MP does a route lookup on the packet for that VRF. The route look points to GRE tunnel as next hop. Control packets, such as ping and routing protocol packets for a VRF, will be encapsulated by the GRE and sent across the GRE tunnel, and sent to the LP for transmission out of the port.

LP CPU forwarding

When the incoming IP packet is more than 1476 bytes (in the default IP MTU scenario) or exceeds the IP MTU of the tunnel interface, the packets must be fragmented and sent with GRE encapsulation. The LP does the fragmentation and sends out the packets. To forward the packets to the correct GRE tunnel as per the VRF of incoming packet, mapping is provided by route entry. This works once the route entry in VRF points to the GRE tunnel as the next hop.

> **NOTE**
> Other tunnel optional configurable parameters for tunnel like Keep alive, TTL, TOS, and so on, are supported by the GRE tunnel.

GRE tunnel VRF limitations

- The GRE tunnel VRF supports only the IPv4 addresses.
- Multicast is not supported on GRE tunnel.
- There is no dynamic CAM model for the IP GRE.
- GRE encapsulation of MPLS packet is also not supported.
- The GRE tunnel VRF support is applicable to all Gen 2 cards except BR-MLX-10Gx24-DM.
- ISIS is not supported for interface having VRF configuration. Hence only static, OSPF, BGP and RIP protocols are supported.
- PBR does not support VRF in current release. However, if we apply a PBR policy to an interface with VRF configured, then PBR will not work, but PBR policies' next-hop can be a tunnel interface irrespective of the tunnel being in any VRF.
- CES/CER does not support VRF over GRE tunnel.

Following **show** commands display the following VRF information:

```
device(config)#show interface tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source  36.0.8.108
  Tunnel destination is 131.108.5.2
  Tunnel mode gre ip
  No port name
  Internet address is: 10.10.3.1/24
  Tunnel TOS 0, Tunnel TTL 255, Tunnel MTU 1476 bytes
  Keepalive is Enabled : Interval 10, No.of Retries 3
  Total Keepalive Pkts Tx: 2, Rx: 2
  VRF Forwarding: Red

device(config)#show ip interface tunnel 1
Interface Tunnel 1
  port enabled
  port state: UP
  ip address: 10.10.3.1/24
  Port belongs to VRF: red
  encapsulation: ETHERNET, mtu: 1476
  directed-broadcast-forwarding: disabled
  ip icmp redirect: enabled
  ip local proxy arp: disabled
```

```
  ip ignore gratuitous arp: disabled
  No inbound ip access-list is set
  No outbound ip access-list is set
  No Helper Addresses are configured.

device(config)# show ip-tunnels 1
IPv4 tnnl 1 UP : src_ip 36.0.8.108, dst_ip 131.108.5.2
TTL 255, TOS 0, NHT 0, MTU 1480, vrf: red
```

Multicast over GRE tunnel

NOTE

MTU fragmentation for multicast traffic is not enabled over a GRE tunnel. Packets are transmitted without MTU fragmentation. This behavior is applicable on Brocade NetIron MLX Series, Brocade NetIron XMR Series, Brocade NetIron CER Series, and Brocade NetIron CES Series devices.

Multi-Service IronWare software supports Multicast over a point-to-point GRE tunnel. Multicast over a GRE tunnel allows multicast packets to be transported through a GRE tunnel across an IP cloud towards its receiver. A GRE tunnel is provisioned at each end of the IP cloud. A GRE tunnel is a virtual IP tunnel; the IP tunnel source can also be a VE interface. The IP cloud sitting in between the two GRE endpoints serves as a PIM enabled logical link. As bidirectional control messages are sent over the GRE tunnel, the multicast distribution tree is established across the IP cloud. Multicast data is encapsulated with a predefined GRE header at the ingress node. The GRE packet is routed within the IP cloud using the outer unicast GRE destination address. As the packet reaches the egress node of the tunnel, the packet is decapsulated. The multicast packet continues on its way to the multicast distribution tree to reach its receivers.

Configuring PIM GRE tunnel

The Brocade device PIM GRE tunnel configuration allows you to enable PIM Sparse (PIM-SM) and PIM Dense (PIM-DM) on a GRE tunnel.

Enabling PIM-SM on a GRE tunnel interface

To enable PIM-SM on a GRE Tunnel Interface, enter the following command.

```
device(config)#interface tunnel 20
device(config-tnif-20)#ip pim-sparse
```

Syntax: [no] ip pim-sparse

Enabling PIM-DM on a GRE tunnel interface

To enable PIM-DM on a GRE Tunnel Interface, enter the following command.

```
device(config)#interface tunnel 20
device(config-tnif-20)#ip pim
```

Syntax: [no] ip pim

Configuring PIM GRE tunnel using the strict RPF check

The device PIM GRE tunnel configuration allows you to enforce strict rpf check rules on (s,g) entry on a GRE tunnel interface. The (s,g) entry uses the GRE tunnel as a RPF interface. During unicast routing transit, GRE tunnel packets may arrive at different physical interfaces. The **ip pim tunnel rpf-strict** command allows you to limit a specific port to accept the (s,g) GRE tunnel traffic.

NOTE
The configuration is not recommended for all users, it is only needed if the user wants to override the default behavior.

When the GRE encapsulated multicast packet is received, hardware processing attempts to find a match in the CAM session based on the inner (s,g) entry. If hardware processing cannot find the inner (s,g) entry in the CAM session, the packet will be dropped. If the **ip pim tunnel rpf-strict** command is configured on a GRE tunnel interface, hardware processing will check on the (s,g) entry, and verify that the packet matches the physical port on the GRE tunnel interface, and the GRE tunnel vlan id.

To limit a specific port to accept the (s,g) GRE tunnel traffic, enter the following command.

```
device(config)#interface tunnel 20
device(config-tnif-20)#ip pim tunnel rpf-strict
```

Syntax: [no] ip pim tunnel [**rpf-strict**]

The rpf-strict option allows you to set the strict rpf check on the multicast entry.

Tunnel statistics for a GRE tunnel or IPv6 manual tunnel

At a global level, you can enable the collection of statistics for generic routing encapsulation (GRE) tunnels and manual IPv6 tunnels. With this feature, the Brocade device collects the statistics for GRE and IPv6 manual tunnels and displays packet counters for tunnels at the management processor (MP). This feature collects and displays unicast and multicast packets over both directions of the tunnels.

Statistics collection is not enabled by default, so you need to enter the IP tunnel policy configuration level and then issue the **accounting-enable** command to start collecting the statistics for GRE and IPv6 manual tunnels. This procedure is described in Enabling tunnel statistics on page 691. This required preliminary ensures that the source-ingress CAM partition is not allocated unless statistics collection or tunnel session enforcement checks are actually needed. (Because the statistics enable does not enforce the GRE and IPv6 tunnel session checks by default, these capabilities have their own enable commands in the IP tunnel policy CLI level. The applicable commands are described in Configuring IPv6 session enforce check on page 692 and Configuring IPv6 session enforce check on page 692.) You can view examples of related show command output in Displaying GRE and manual IPv6 tunnel statistics on page 803.

The remainder of this introduction to tunnel statistics describes reload behavior for certain commands and detailed notes and restrictions that apply to the support for tunnel statistics.

Reload behavior and the source-ingress CAM partition

When one of the three tunnel-related commands is configured at the CLI level for IP tunnel policy (entered by use of the **ip-tunnel-policy** command), you might need to save the configuration and reload the device to create the required source-ingress-CAM partition. If the memory write and reload are needed, the system prompts for these steps after you finish the enable commands. The condition for which you might need to write and reload is the absence of the source-ingress-CAM partition. If this partition does not exist, the first time that you run either the **gre-session-enforce-check** , **ipv6-session-enforce-check** , or **accounting-enable** command, the system prompts you. Thereafter, when you run any of these three commands to disable or enable a feature, the system does not prompt. Removing any of the configurations can be done at anytime and does not necessitate a reload. The new configuration immediately becomes effective, but the source-ingress CAM partition is removed only upon the next reload.

Operational notes

The subsections that following describe operational characters that relate to the statistics collection.

Source-ingress-CAM partition

The CAM profile restrictions for this feature are the same as those for the tunnel session enforce-check configuration. This feature is not supported in those CAM profiles for which the system cannot allocate the source-ingress-CAM partition that is needed to support the accounting and session check enforcement. The CLI engine checks for compliance and rejects an attempt to enable statistics in this situation. Currently the following CAM profiles are not supported for IP tunnel statistics:

- IPv6
- L2-metro-2
- MPLS-L3VPN-2
- MPLS-VPLS-2
- MPLS-VPN-VPLS
- multi-service-2
- multi-service-3
- multi-service-4

6to4 automatic tunnels

Statistics collection is supported only for manual IPv6 and GRE tunnels. The system does not support statistics collection for 6to4 automatic IPv6 tunnels because, for automatic 6to4 tunnels, only tunnel source-ip is configured, and the destination is known only at runtime when a remote node tries to use this tunnel. The destination points can come up or go down without the local router having any information on how many destinations are to be used for 6to4 tunnels. This uncertainty can cause scalability issues, so neither statistics collection nor session-enforce check are not supported for 6to4 automatic tunnels.

Multicast-over-GRE packets

This feature counts multicast over GRE packets. You can see the multicast packet count by using the **show interface tunnel** command. You can use other CLI commands to display the aggregate unicast and multicast statistics for the GRE tunnels. For a description of all the applicable show commands, refer to Displaying GRE and manual IPv6 tunnel statistics on page 803.

Statistics polling on the MP and LP

The LP module polls the statistics once every second. For every one second, the module polls the statistics either for 5000 entries or until the completion of a specific application. (The same polling mechanism is also used for other applications, such as IP, MPLS, L3VPN, VLL, VPLS and IP Tunnel.) After all the applications are polled, the system waits for 220 seconds to schedule the next polling event. However, the LP module synchronizes statistics to the MP every 30 seconds, so 30 seconds is the granularity of statistics.

The LP synchronizes statistics to the MP in background every 30 seconds, and the MP stores the statistics for all tunnels for every LP module. If a LP module at either the tunnel ingress or egress, the system uses the current stored statistics for that LP module for display (and continue to poll the rest of working modules to get the latest statistics). This mechanism ensures that the tunnel counters never go down (if no clear statistics command is performed on the tunnel).

When a tunnel is down, the LP does not poll the statistics for that tunnel. The LP keeps the old counters as is until you explicitly clear them on the CLI. These counters are displayed when the tunnel is down. When the tunnel comes back up, it resumes polling and adds the new packet counts to the stored statistics and displays the updated statistics.

Clearing the statistics

When you issue the **clear statistics tunnel** command with specific parameters, the operation clears statistics for either one or all of the tunnels regardless of the circumstance-- whether the tunnel is up or down, on an ingress or egress module, and so on. Refer to Clearing GRE tunnel and manual IPv6 tunnel statistics on page 692 for a description of the clear statistics tunnel command.

Tunneled packets that encounter an ACL

If a packet reaches the ACL permit or deny clauses for the inner IPv4 or IPv6 addresses when it comes through the IP tunnel at the egress node, the packet is not counted as a receive-from-tunnel packet. Instead, it is counted as an ACL packet. You can view ACL packets by using the show access-list accounting command.

IPv6 ACL lookup is performed on the inner IPv6 packet at the tunnel egress. This depends on the port register for the Layer 2 ACL or Layer 3 ACL control, which is performed in parallel.

Switchover behavior

The LP sends statistics to both the active and the standby MP modules. If an MP switches over, the new-active MP polls the statistics again so it can display the latest statistics. The counters are equal to or greater than the statistics before the switchover for the working modules. If any module goes down before the switchover, the new active MP uses the stored counters to display the statistics for that module.

Hitless operating system upgrade behavior

When a hitless operating system (OS) upgrade occurs, the tunnel statistics are saved and retrieved after the reset of the LP is complete. The system can retrieve the old statistics and do the polling to get the latest PRAM statistics. After the hitless upgrade, the system can display the correct packet counters.

Behavior after an LP failure

If LP module goes down, the counters for that LP are preserved. After the LP comes back up, the preserved counters for that LP can be displayed.

Feature scalability

A Brocade NetIron XMR Series device supports 256 tunnels by default and 8000 tunnels for its maximum number of tunnels. The system supports statistics for all tunnels because the source ingress CAM partition has 16000 entries that can support the statistics for all tunnels.

Enabling IP tunnel or manual IPv6 statistics

This section describes how to enable and clear statistics for GRE or manual IPv6 tunnels. The enable for this feature is global in scope. The enabling command is one of three enable commands that you run in the IP tunnel policy context of the CLI. (These commands are **gre-session-enforce-check** , **ipv6-session-enforce-check** , and **accounting-enable** . The **ip-tunnel-policy** command puts the CLI in the mode for executing them.) To see examples of tunnel statistics, refer to Displaying GRE and manual IPv6 tunnel statistics on page 803.

Enabling tunnel statistics

> NOTE
> The Brocade NetIron CES Series and Brocade NetIron CER Series devices currently do not support the **ip-tunnel-policy** and the **accounting-enable commands.**

To enable the GRE tunnel or manual IPv6 tunnel statistics, go to the IP tunnel policy mode of the CLI and issue the **accounting-enable** command, as the following example illustrates.

```
device(config)#ip-tunnel-policy
device(config-ip-tunnel-policy)#accounting-enable
```

Syntax: [no] accounting-enable

To turn off tunnel statistics gathering, use the keyword **no** to the accounting-enable command.

The system might prompt you to write the configuration to memory and reload the system. If the system has not yet allocated a source-ingress CAM partition, it prompts you to write the results of the current configuration to memory and reload the system.

The first-time execution of certain commands can prompt the allocation of a source-ingress CAM partition that is required by certain features. These commands are **gre-session-enforce-check** , **ipv6-session-enforce-check** , and **accounting-enable** . After this CAM partition is allocated, you do not need to do the memory write and reload after the first-time execution of the other two commands.

Clearing GRE tunnel and manual IPv6 tunnel statistics

You can clear all of the statistics for either one or all tunnels by using the clear statistics tunnel command, as the following example illustrates.

```
device#clear statistics tunnel 1
```

Syntax: clear statistics tunnel [**tunnel ID**]

To clear statistics for a specific tunnel, include the ID of that tunnel.

Configuring IPv6 session enforce check

You can enable the IPv6 session enforce check by using the **ipv6-session-enforce-check** command. When an IPv6 packet arrives and this feature is enabled, the system tries to match the IPv6 packet source and destination address pair with the tunnel configured destination and source pair. If the pairs do not match, the packet is dropped in hardware.

> NOTE
> The Brocade NetIron CES Series and Brocade NetIron CER Series devices currently do not support the **ip-tunnel-policy** command or IPv6 **tunnels.**

To configure the IPv6 session enforce check, go to the IP tunnel policy context and enter the **ipv6-sessionenforce-check** command.

```
device(config)#ip-tunnel-policy
device(config-ip-tunnel-policy)#ipv6-session-enforce-check
```

Syntax: [no] ipv6-session-enforce-check

To disable the IPv6 session enforce check, use the no form of this command.

The system might prompt you to write the configuration to memory and reload the system. If the system has not yet allocated a source-ingress CAM partition, it prompts you to write the results of the current configuration to memory and reload the system.

The first-time execution of certain commands can prompt the allocation of a source-ingress CAM partition that is required by certain features. These commands are gre-session-enforce-check, ipv6-session-enforce-check, and accounting-enable. After this CAM partition is allocated, you do not need to do the memory write and reload after the first-time execution of the other two commands.

> NOTE
> The **ipv6-sessions-enforce-check** command is not supported for 6to4 automatic tunnels.

Restart global timers

Restart contains two global timers that:

- Limit the amount of time used for re-syncing routes between the backup Management module and Interface modules (LPs) within the same chassis
- Allow a buffer time for protocols to converge and solve dependencies among each other

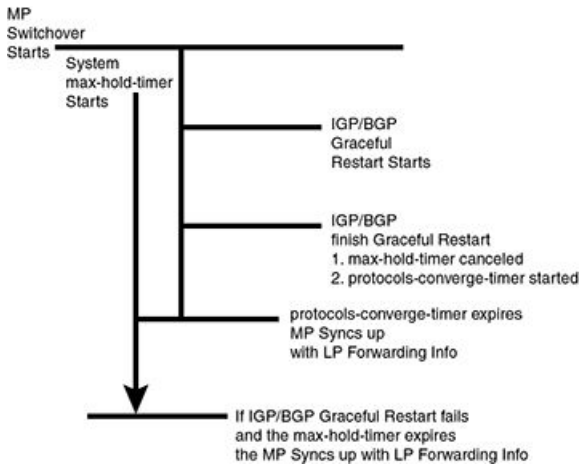
If the protocol-based restart features are configured when a Management module (MP) performs a switchover to the its backup, routes are maintained on the LPs through the protocol-based restart processes for a specified period of time while the new MP learns the network routes. Once the MP learns all of its routes, the routes from the MP are synced with the routes on the LPs.

The two timers introduced here are called the **max-hold-timer** and the **protocols-converge-timer** .

The process of syncing routes between a new MP and its LPs using the new timers are illustrated in Figure 186 and described in the following steps.

1. The MP switchover from active to redundant MP begins.
2. The system **max-hold- timer** starts.
3. The IGP/BGP restart process begins.
4. If the IGP/BGP restart process is completed before the system **max-hold-timer** expires, the system **max-hold-timer** is cancelled and the **protocols-converge-timer** starts.
5. Once the **protocols-converge-timer** expires, the MP syncs up forwarding information with the LPs.
6. If the system **max-hold-timer** expires before the IGP/BGP restart process is completed, the MP syncs up forwarding information with the LPs at that time and the **protocols-converge-timer** is never started.

FIGURE 183 MP to LP re-syncing process



Configuring the graceful-restart max-hold-timer

This timer defines the maximum hold time before a management module syncs up new forwarding information to interface modules during the restart process. While the default value of 300 seconds will work in most cases, if a device is loaded with a very large number of routes and OSPF/BGP peering adjacencies you might want to fine-tune your device's performance by increasing this value.

The value of this timer can be set using the command shown in the following.

```
device(config)# graceful-restart max-hold-timer 500
```

Syntax: [no]graceful-restart max-hold-timer hold-time

The *hold-time* variable is the maximum number of seconds that a management routing module waits before it syncs up new forwarding information to the interface modules during a restart. The range for the hold time is 30 - 3600 seconds. The default time is 300 seconds.

Graceful-restart protocols-converge-timer

This timer defines the time that a Brocade device waits for restarting protocols to converge at the final step in the restart process. In a heavily loaded system where BGP/OSPF/GRE/Static protocols can have a dependency on each other, their restart procedures may also depend on each other. This timer is to allow protocols to solve inter-dependencies after individual restart processes and before routing modules sync up new forwarding information to interface module. The default value of 5 seconds will work in most cases but if a system is heavily loaded and has protocols that depend on each other, you might want to fine-tune your system by increasing this value.

The value of this timer can be set using the command shown in the following.

```
device(config)# graceful-restart protocols-converge-timer 20
```

Syntax: [no]graceful-restart protocols-converge-timer hold-time

The *hold-time* variable is the maximum hold time in seconds before management routing modules sync up new forwarding information to interface modules during restart. The range of permissible values is 0 to 1200 seconds. The default value is 5 seconds.

Configuring IP parameters

Some parameters can be configured globally while others can be configured on individual interfaces. Some parameters can be configured globally and overridden for individual interfaces.

Configuring IP addresses

You can configure an IP address on the following types of the Brocade device interfaces:

- Ethernet port
- Virtual routing interface (also called a Virtual Ethernet or "VE")
- Loopback interface

By default, you can configure up to 24 IP addresses on each interface.

> NOTE
> After you configure a virtual routing interface on a VLAN, you cannot configure Layer 3 interface parameters on individual ports in the VLAN. Instead, you must configure the parameters on the virtual routing interface itself. Also, after an IP address is configured on an interface, the hardware is programmed to route all IP packets that are received on the interface. Consequently, all IP packets not destined for this device's MAC address are not bridged and are dropped.

The Brocade device supports both classical IP network masks (Class A, B, and C subnet masks, and so on) and Classless Interdomain Routing (CIDR) network prefix masks.

- To enter a classical network mask, enter the mask in IP address format. For example, enter "10.157.22.99 255.255.255.0" 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 "10.157.22.99/24" for an IP address that has a network mask with 24 significant bits (ones).

By default, the CLI displays network masks in classical IP address format (example: 255.255.255.0). You can change the display to prefix format.

Assigning an IP address to an Ethernet port

To assign an IP address to port 1/1, enter the following commands.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip address 10.45.6.1 255.255.255.0
```

NOTE
You also can enter the IP address and mask in CIDR format, as follows.

```
device(config-if-e10000-1/1)# ip address 10.45.6.1/24
```

Syntax: [no] interface ethernet slot/port

Syntax: [no] ip address ip-addr ip-mask | ip-addr/mask-bits [ospf-ignore | ospf-passive | secondary]

The **ospf-ignore** and **ospf-passive** parameters modify the Brocade device defaults for adjacency formation and interface advertisement. Use one of these parameters if you are configuring multiple IP subnet addresses on the interface but you want to prevent OSPF from running on some of the subnets:

- **ospf-passive** - disables adjacency formation with OSPF neighbors (but does not disable advertisement of the interface into OSPF). By default, when OSPF is enabled on an interface, the software forms OSPF router adjacencies between each primary IP address on the interface and the OSPF neighbor attached to the interface.
- **ospf-ignore** - disables OSPF adjacency formation and advertisement of the interface into OSPF. The subnet is completely ignored by OSPF.

Use the **secondary** parameter if you have already configured an IP address within the same subnet on the interface.

NOTE
When you configure more than one address in the same subnet, all but the first address are secondary addresses and do not form OSPF adjacencies.

Assigning an IP address to a loopback interface

Loopback interfaces are always up, regardless of the states of physical interfaces. They can add stability to the network because they are not subject to route flap problems that can occur due to unstable links between a Brocade device and other devices.

You can configure up to 64 loopback interfaces on a Brocade device.

You can add up to 24 IP addresses to each loopback interface.

NOTE
If you configure the Brocade device to use a loopback interface to communicate with a BGP4 neighbor, you also must configure a loopback interface on the neighbor and configure the neighbor to use that loopback interface to communicate with the Brocade device.

To add a loopback interface, enter commands such as those shown in the following example.

```
device(config-bgp-router)# exit
device(config)# int loopback 1
device(config-lbif-1)# ip address 10.0.0.1/24
```

Syntax: [no] interface loopback num

For the syntax of the IP address, refer to Assigning an IP address to an Ethernet port on page 694.

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 Brocade device.

NOTE
Other sections in this chapter that describe how to configure interface parameters also apply to virtual interfaces.

NOTE
The Brocade device uses the lowest MAC address on the device (the MAC address of port 1 or 1/1) as the MAC address for all ports within all virtual interfaces you configure on the device.

To add a virtual interface to a VLAN and configure an IP address on the interface, enter commands such as the following.

```
device(config)# vlan 2 name IP-Subnet_10.1.2.0/24
device(config-vlan-2)# untag e1/1 to 1/4
device(config-vlan-2)# router-interface ve1
device(config-vlan-2)# interface ve1
device(config-vif-1)# ip address 10.1.2.1/24
```

The first two commands create a Layer 3 protocol-based VLAN named "IP-Subnet_1.1.2.0/24" and add a range of untagged ports to the VLAN. The **router-interface** command creates virtual interface 1 as the routing interface for the VLAN. The last two commands change to the interface configuration level for the virtual interface and assign an IP address to the interface.

Syntax: [no] router-interface ve num

Syntax: [no] interface ve num

The *num* parameter specifies the virtual interface number. You can specify from 1 to the maximum number of virtual interfaces supported on the device. To display the maximum number of virtual interfaces supported on the device, enter the **show default values** command. The maximum is listed in the System Parameters section, in the Current column of the virtual-interface row.

For the syntax of the IP address, refer to Assigning an IP address to an Ethernet port on page 694.

Assigning a MAC address to a virtual interface

By default, the Brocade device uses the MAC address of the first port (1 or 1/1) as the MAC address for all virtual routing interfaces configured on the device. You can specify a different MAC address for the virtual routing interfaces. If you specify another MAC address for the virtual routing interfaces, the address applies to all the virtual routing interfaces configured on the device. To specify the MAC address for virtual routing interfaces, enter commands such as the following.

```
device(config)# virtual-interface-mac aaaa.bbbb.cccc
device(config)# write memory
device(config)# end
device# reload
```

Syntax: [no] virtual-interface-mac mac-addr

Enter the MAC address in the following format: HHHH.HHHH.HHHH

NOTE
You must save the configuration and reload the software to place the change into effect.

Deleting an IP address

To delete an IP address, enter a command such as the following.

```
device(config-if-e1000-1/1)# no ip address 10.1.2.1
```

This command deletes IP address 10.1.2.1. You do not need to enter the subnet mask.

To delete all IP addresses from an interface, enter the following command.

```
device(config-if-e1000-1/1)# no ip address *
```

Syntax: no ip address ip-addr

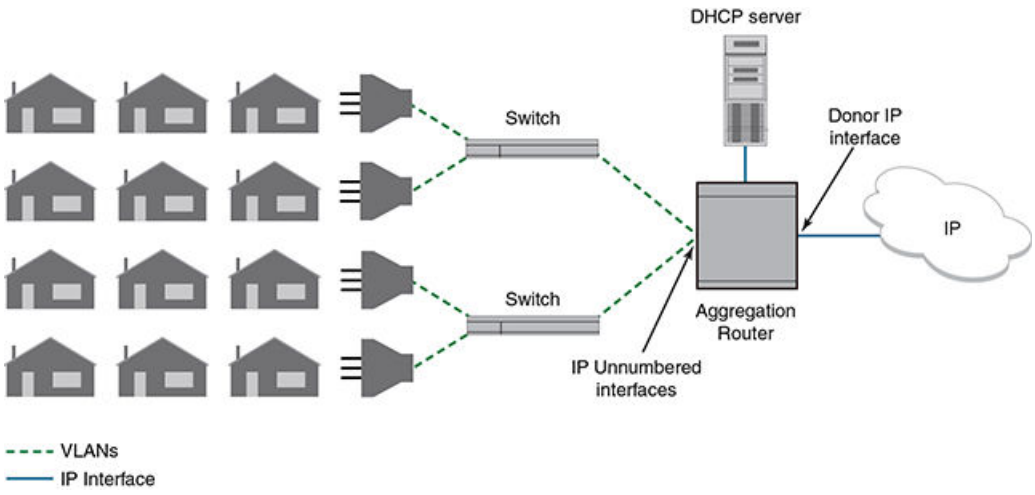
IP Unnumbered Interfaces

The IP Unnumbered Interfaces feature saves IPv4 address space by allowing unnumbered interfaces to inherit the IP address of a donor interface, thus allowing all ports to share the same subnet. This feature not only preserves IP addresses, but also reduces the IP routing table size. This feature also provides ARP suppression (reducing the number of ARP requests sent to hosts) on unnumbered interfaces, thus increasing the number of hosts that are supported under the same subnet.

- The *donor interface* is the interface with an IP address configured on it.
- The *unnumbered interface* is the interface with no IP address configured on it. The unnumbered interface inherits the IP address of the donor interface.

For example, consider a DSLAM deployment scenario with multiple users connected to a Brocade device (refer to Figure 187). Instead of configuring IP addresses for every VE on the Brocade device, you can designate one VE as the donor interface and configure all the other VEs to inherit the IP address of the donor VE interface.

FIGURE 184 IP Unnumbered Interfaces feature



The donor interface must be one of the following:

- Loopback interface
- VE interface
- Ethernet interface (can be part of a LAG interface; must be untagged if in a VLAN)

The unnumbered interfaces can be the following:

- VE interface
- Ethernet interface (must be untagged if in a VLAN)

Configuring an unnumbered interface

To enable an unnumbered interface to inherit the IP address of a donor interface, enter commands such as the following:

```
device (config)# interface ve 10
device (config-vif-10)# ip unnumbered ve 9
```

The commands enable interface ve 10 to inherit the IP address of ve 9. Interface ve 10 is the unnumbered interface and interface ve 9 is the donor interface.

Syntax: [no] ip unnumbered [**ethernet** *slot/port* | **ve** *num* | **loopback** *num*]

The **ethernet***slot/port* parameter specifies the donor interface by an Ethernet port number.

The **ve***num* parameter specifies the donor interface by virtual interface number.

The **loopback***num* parameter specifies the donor interface by loopback interface number.

Use the **no ip unnumbered** command to remove the IP address from the unnumbered interface.

NOTE
You do not need to configure an interface to be a donor interface. An interface becomes a donor interface automatically when an unnumbered interface inherits its IP address.

Displaying unnumbered interfaces

The **show ip interface** command displays information about unnumbered interfaces.

In the following example, note that interfaces ve 9 and ve 10 have the same IP address. Interface ve 10 is an unnumbered interface, as indicated by the **U** in the **Flag** column.

```
device# show ip interface

Interface       IP-Address      OK?  Method Status    Protocol VRF           FLAG
mgmt 1          10.21.108.35    YES  NVRAM  up        up       default-vrf
ve 6            6.1.1.1         YES  NVRAM  up        up       default-vrf
ve 9            1.1.1.1         YES  NVRAM  up        up       default-vrf
ve 10           1.1.1.1         YES  NVRAM  up        up       default-vrf   U
```

In the following example, the first highlighted line indicates that interface ve 10 is an unnumbered interface, inheriting the IP address of ve 9, which is the donor interface.

```
device# show ip interface ve 10
Interface Ve 10
  members: ethe 4/1
  active: ethe 4/1
  port enabled
  port state: UP
  ip address: 1.1.1.1/24
  Unnumbered interface, Using IP address of  ve 9
  unnumbered arp-suppression is enabled
  Port belongs to VRF: default-vrf
(output truncated)
```

The second highlighted line indicates whether ARP suppression is enabled or disabled. Refer to ARP suppression on unnumbered interfaces on page 698 for information about ARP suppression.

ARP suppression on unnumbered interfaces

When you configure unnumbered interfaces, those interfaces are configured with ARP suppression by default. This means that ARP requests are not sent out on unnumbered interfaces. If many VLANs belong to the same subnet, this avoids an ARP storm.

Donor interfaces continue to send out ARP requests because, by default, ARP suppression is disabled on donor interfaces.

ARP suppression is achieved by enabling ARP suppression and performing one of the following:

- Configure DHCP option 82 (recommended).

This configuration must be enabled on each VLAN belonging to the donor or unnumbered interface. When DHCP option 82 is enabled, the ARP request is sent only to the corresponding VLAN (identified in the Dynamic ARP Inspection (DAI) table) instead of all the unnumbered VLANs. For details of DHCP option 82 and the DAI table, refer to DHCP option 82 insertion on page 729.

- Configure static DAI entries.

You must configure static DAI entries for scenarios where the host is not discovered through DHCP, such as when a host is provided with a static IP address. Refer to Configuring DAI on page 719 for instructions.

> NOTE
> If you enable ARP suppression on a donor interface, you must configure static Dynamic ARP Inspection (DAI) entries for protocol neighbor IP addresses to ensure that protocol operations on the donor interface succeed.

Refer to How ARP works on page 713 for information about ARP requests.

Enabling and disabling ARP suppression

To enable or disable ARP suppression on an unnumbered or donor interface, enter commands such as the following:

```
device (config)# interface ve 9
device (config-vif-9)# ip unnumbered-arp-suppression
device (config)# interface ve 10
device (config-vif-10)# no ip unnumbered-arp-suppression
```

The commands enable ARP suppression on ve 9 and disable ARP suppression on ve 10. To fully achieve ARP suppression, configure one of the following:

- DHCP option 82 (refer to Enabling DHCP snooping on a VLAN on page 727)
- Static DAI entries (refer to Configuring DAI on page 719)

Syntax: [no] ip unnumbered-arp-suppression

Use the **no ip unnumbered-arp-suppression** command to disable ARP suppression on the interface.

This command is applicable only to donor and unnumbered interfaces. It has no effect on other interfaces.

You can use the **show ip interface** command to display whether ARP suppression is enabled or disabled, as shown in Displaying unnumbered interfaces on page 698.

Caveats and limitations for IP Unnumbered Interfaces

- The IP Unnumbered Interfaces feature is not supported for IPv6 addresses.
- Multicast and MPLS protocols are not supported on donor interfaces.
- Routing protocols, multicast, and MPLS are not supported on the unnumbered interfaces.
- The IP Unnumbered Interfaces feature is supported only on the default VRF. Both donor and the unnumbered interfaces must be in the default VRF.
- If the donor interface is down (link state or administrative state), a ping to the donor IP address fails, even if the unnumbered interfaces that inherited the IP address are up.
- VRRP and VRRP-E operations are not supported on unnumbered interfaces.
- RPF strict mode is not supported on unnumbered interfaces.

Configuration considerations for IP Unnumbered Interfaces

- You can have multiple donor interfaces in the device. A donor interface can have multiple unnumbered interfaces inheriting its IP address. An unnumbered interface can have only one donor interface.
- You can configure multiple primary and multiple secondary IP addresses on the donor interface. The unnumbered interface inherits all primary and secondary addresses of the donor interface.
- The unnumbered interface inherits only the IP address from the donor interface. All other donor interface configurations are not passed on to the unnumbered interface. You must configure other features, such as IP Source Guard and forwarding of directed broadcasts, on the unnumbered interfaces separately.
- The following routing protocols are supported on the donor interface:
  - Open Shortest Path First (OSPF)
  - Intermediate System - Intermediate System (IS-IS)
  - Routing Information Protocol (RIP)
  - Border Gateway Protocol (BGP)
- If DHCP clients are configured on an unnumbered interface, then DHCP option 82 must be configured on that interface; otherwise, the DHCP client cannot get the IP address from the DHCP server.
- If reachability is needed between two hosts within the same subnet, you must configure local proxy ARP on the unnumbered interfaces. Refer to Enabling local proxy ARP on page 716 for more information.

Static route considerations for unnumbered interfaces

- If you configure a static route with an unnumbered interface or donor interface as the next hop, it is recommended that you configure a standard static route instead of an interface-based static route.
- If you configure an interface-based static route on a donor or unnumbered interface, you must ensure that ARP suppression is disabled on the interface. Refer to Enabling and disabling ARP suppression on page 699 for instructions.

Refer to Static route types on page 746 and Configuring a static IP route on page 747 for information about static routes.

DHCP host subnet selection

If the donor interface is configured with multiple subnets, and the DHCP clients need to receive addresses in a specific subnet, use the **ip bootp-gateway** command to select the local donor interface IP address of the specific subnet.

This functionality can be used when the DHCP clients are moved from one subnet to another subnet.

Refer to Changing the IP address used for stamping BootP or DHCP requests on page 779 for instructions on using the **ip bootp-gateway** command. Note that the **ip bootp-gateway** command is used only when the hosts are DHCP hosts.

Support for other features

IP address configurations are the only configurations that the unnumbered interfaces inherit from the donor interface.

All other configurations (such as ICMP, ACLs, DHCP, and PBR) that are configured on the donor interface apply only to the donor interface and are not inherited by the unnumbered interfaces. You must configure these features separately on the unnumbered interfaces.

Sample configuration for IP Unnumbered Interfaces

This example shows how to configure IP unnumbered interfaces with a DHCP server. In this example, loopback 1 is the donor interface, and ve 20 and ve 30 are the unnumbered interfaces.

First, configure an IP address on the donor interface. Then configure the two VE interfaces to inherit the IP address of the donor interface.

```
device (config)# interface loopback 1
device (config-lbif-1)# ip address 10.10.10.1/24
device (config-lbif-1)# vlan 20
device (config-vlan-20)# router-interface ve 20
device (config-vlan-20)# interface ve 20
device (config-vif-20)# ip unnumbered loopback 1
device (config-vif-20)# vlan 30
device (config-vlan-30)# router-interface ve 30
device (config-vlan-30)# interface ve 30
device (config-vif-30)# ip unnumbered loopback 1
```

Configure the DHCP server. In this example, the DHCP server address is 10.40.40.4.

```
device (config-vif-30)# interface ethernet 1/2
device (config-if-e1000-1/2)# ip address 10.40.40.1/24
device (config-if-e1000-1/2)# dhcp-snooping-trust
```

Configure the DHCP server address in the unnumbered interfaces.

```
device (config-if-e1000-1/2)# interface ve 20
device (config-vif-20)# ip helper-address 10.40.40.4
device (config-vif-20)# interface ve 30
device (config-vif-30)# ip helper-address 10.40.40.4
device (config-vif-30)# exit
```

Configure DHCP option 82 in the unnumbered interface VLANs.

```
device (config)# ip dhcp-snooping vlan 20 to 30 insert-relay-information
```

Support for a 31-bit subnet mask on point-to-point networks

NOTE

The configuration of an IPv4 address with a 31-bit subnet mask is supported on Brocade NetIron MLX Series, Brocade NetIron XMR Series, and Brocade NetIron CER Series and Brocade NetIron CES Series devices.

In an effort to conserve IPv4 address space, a 31-bit subnet mask can be assigned to point-to-point networks. Support for an IPv4 address with a 31-bit subnet mask is described in RFC 3021. Previously, four IP addresses with a 30-bit subnet mask were allocated on point-to-point networks. A 31-bit subnet mask uses only two IP addresses; all zero bits and all one bits in the host portion of the IP address. The two IP addresses are interpreted as host addresses, and do not require broadcast support because any packet that is transmitted by one host is always received by the other host at the receiving end. Therefore, directed broadcast on a point-to-point interface is eliminated. Also, a broadcast address with all one bits in the host portion of the IP address is not allocated for point-to-point interface configuration.

NOTE

IP-directed broadcast CLI configuration at the global level, or the per- interface level, is not applicable on interfaces configured with a 31-bit subnet mask IP address.

Configuring an IPv4 address with a 31-bit subnet mask

To configure an IPv4 address with a 31-bit subnet mask, enter the following commands.

NOTE

You can configure an IPv4 address with a 31-bit subnet mask on any interface (for example, Ethernet, loopback, VE, or tunnel interfaces), and on all VRFs (default and non-default VRFs).

```
device(config)# interface ethernet 1/5
device(config-if-e1000-1/5)# ip address 10.9.9.9 255.255.255.254
```

You can also enter the IP address and mask in the Classless Interdomain Routing (CIDR) format, as follows.

```
device(config-if-e1000-1/5)# ip address 10.9.9.9/31
```

Syntax: [no] ip address *ip-addressip-mask*

Syntax: [no] ip address *ip-address/subnetmask-bits*

The *ip-address* variable specifies the host address. The *ip-mask* variable specifies the IP network mask. The *subnet mask-bits* variable specifies the network prefix mask.

To disable configuration for an IPv4 address with a 31-bit subnet mask on any interface, use the **no** form of the command.

You cannot configure a secondary IPv4 address with a 31-bit subnet mask on any interface. The following error message is displayed when a secondary IPv4 address with a 31-bit subnet mask is configured.

```
device(config-if-e1000-1/5)#ip address 10.8.8.8/31 secondary
IP/Port: Errno(10) Cannot assign /31 subnet address as secondary
```

Displaying the configuration for an IPv4 address with a 31-bit subnet mask

To display the interface running configuration when an IPv4 address with a 31-bit subnet mask is configured, enter the following command at any level of the CLI.

```
device(config-if-e1000-1/5)# show run interface ethernet 1/5
interface ethernet 1/5
enable
ip address 10.2.2.3/31
ip address 10.4.4.4/31
```

In the previous example, interface ethernet 1/5 is assigned two IPv4 addresses (10.2.2.3/31 and 10.4.4.4/31) with a 31-bit subnet mask.

To display the configuration for an IPv4 address with a 31-bit subnet mask on a virtual ethernet (VE) interface, enter the following command at any level of the CLI. In the example below, VE interface 10 is assigned two IPv4 addresses (10.25.25.255/31 and 10.168.32.0/31) with a 31-bit subnet mask.

```
device(config-if-e1000-2/5)#show run interface ve 10
interface ve 10
ip ospf area 0
ip address 10.25.25.255/31
ip address 10.168.32.0/31
```

Syntax: show run interface [**ethernet** *slot/port* | **loopback** *number* | **tunnel** *number* | **ve** *number*]

The **show ip route** command displays routes that are directly connected with interfaces configured with IPv4 addresses with a 31-bit subnet mask.

```
device(config-if-e1000-2/5)# show ip route
Total number of IP routes: 21
        Destination        Gateway        Port        Cost     Type  Uptime
1        10.2.2.2/31        DIRECT         eth 1/5     0/0        D    2h19m
2        10.4.4.4/31        DIRECT         eth 1/5     0/0        D    2h19m
3        10.25.25.254/31    DIRECT         ve 10       0/0        D    2h25m
4        10.168.32.0/31     DIRECT         ve 10       0/0        D    2h25m
```

Syntax: show ip route

Enabling hardware forwarding of IP option packets based on Layer 3 destination

The IP option field in an IP header is variable in length. A packet can have zero or more options and an option can have either of the following forms:

- a single octet of option-type
- an option-type octet, an option-length octet, and option-data octets

The option-type octet consists of the following three fields:

- 1 bit copied flag
- 2 bits option class
- 5 bits option number

By default, IP option packets are sent to the CPU for forwarding. When configured on a physical interface, the **ignore-options** command directs the device to ignore all options in IP option packets that are received at the configured port. These packets are then treated as if there were no options configured and forwarded based on their Layer-3 destination. The **ignore-options** command is configured as shown in the following.

```
device(config)# interface ethernet 1/1
device(config-if-e1000-1/1)# ignore-options
```

Syntax: [no] ignore-options

This command only applies to IP option packets in the default VRF.

When the **ignore-options** command is configured on a port, RSVP router alert packets incoming on that port will not be sent to the CPU. Consequently, MPLS should not be configured on a physical port where the **ignore-options** command is configured.

Using the ignore-options command in a LAG configuration

The **ignore-options** command can be used on a LAG but it must apply to all ports on the LAG. This applies to both static and LACP LAGs as described in the following:

Configuring the ignore-options command on a static LAG

To configure the **ignore-options** command on a static LAG, each port on the LAG must be configured with the command. You can do this by configuring the command on each port before the LAG configuration or configuring the **ignore-options** command on the primary port of the LAG which automatically applies the command to all ports on the LAG as shown in the following.

```
device(config)# trunk e 3/1 to 3/4
trunk transaction done.
device(config-trunk-3/1-3/4)# exit
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# ignore-options
```

If the LAG is removed, the **ignore-options** command will be propagated to all ports that were previously in the LAG.

If you try to create a LAG where some of the ports have the **ignore-options** command configured and some do not, the LAG will not be allowed as shown in the following example.

```
device(config)# trunk e 3/1 to 3/2
port 3/1 ignore-options is Enabled, but port 3/2 ignore-options is Disabled
Error: port 3/1 and port 3/2 have different configurations
trunk transaction failed:  trunk Config Vetoed
```

Configuring the ignore-options command on a LACP LAG

Just as with static LAGs, if you want to configure the **ignore-options** command on an LACP LAG, the command must be enabled on all ports within the LAG. If it is not, the LACP LAG will not be accepted as shown in the following.

```
device(config)#lag sta_lag static
device(config-lag-sta_lag)#ports e 1/3 to 1/4
device(config-lag-sta_lag)#primary-port 1/3
device(config-lag-sta_lag)#deploy
device(config-lag-sta_lag)#int e 1/3
device(config-if-e1000-1/3)#ignore-options
device(config)#lag sta_lag static
device(config-lag-sta_lag)#ports e 1/3 e 1/4
device(config-lag-sta_lag)#primary-port 1/3
device(config-lag-sta_lag)#deploy
port 1/3 ignore-options is Enabled, but port 1/4 ignore-options is Disabled
Error: port 1/3 and port 1/4 have different configurations
LAG sta_lag deployment failed!
device(config)#int e 1/3
device(config-if-e1000-1/3)#ignore-options
device(config-if-e1000-1/3)#lag dyn_lag dynamic
device(config-lag-dyn_lag)#ports e 1/3 e 1/4
device(config-lag-dyn_lag)#primary-port 1/3
device(config-lag-dyn_lag)#deploy
port 1/3 ignore-options is Enabled, but port 1/4 ignore-options is Disabled
Error: port 1/3 and port 1/4 have different configurations
LAG dyn_lag deployment failed!
```

Configuring domain name server (DNS) resolver

The DNS resolver lets you use a host name to perform Telnet, ping, and traceroute commands. You can also define a DNS domain on a Brocade device and thereby recognize all hosts within that domain. After you define a domain name, the Brocade device automatically appends the appropriate domain to the host and forwards it to the domain name server.

For example, if the domain "newyork.com" is defined on a Brocade device and you want to initiate a ping to host "NYC01" on that domain, you need to reference only the host name in the command instead of the host name and its domain name. For example, you could enter either of the following commands to initiate the ping.

```
device# ping nyc01
device# ping nyc01.newyork.com
```

Multiple DNS queries can be executed simultaneously, making it possible for the Brocade device to run multiple simultaneous Telnet, ping or traceroute commands using host names.

Defining an IPv4 DNS entry

You can define up to four DNS servers for each DNS entry. The first entry serves as the primary default address. If a query to the primary address fails to be resolved after three attempts, the next gateway address is queried (also up to three times). This process continues for each defined gateway address until the query is resolved. The order in which the default gateway addresses are polled is the same as the order in which you enter them.

Suppose you want to define the domain name of abc.com on a device and then define four possible default DNS gateway addresses. To do so using IPv4 addressing, you would enter the following commands.

```
device(config)# ip dns domain-name abc.com
device(config)# ip dns server-address 10.157.22.199 10.96.7.15 10.95.7.25 10.98.7.15
```

Syntax: [no] ip dns server-address ip-addr [ip-addr] [ip-addr] [ip-addr]

In this example, the first IP address in the **ip dns server-address** command becomes the primary gateway address and all others are secondary addresses. Because IP address 10.98.7.15 is the last address listed, it is also the last address consulted to resolve a query.

DNS queries of IPv4 and IPv6 DNS servers

IPv4 and IPv6 DNS record queries search through IPv4 and IPv6 DNS servers as described in the following:

For IPv4 DNS record queries:

- Loop thru all configured IPv4 DNS servers,
- If no IPv4 DNS servers were configured, then loop through all configured IPv6 DNS servers (if any).

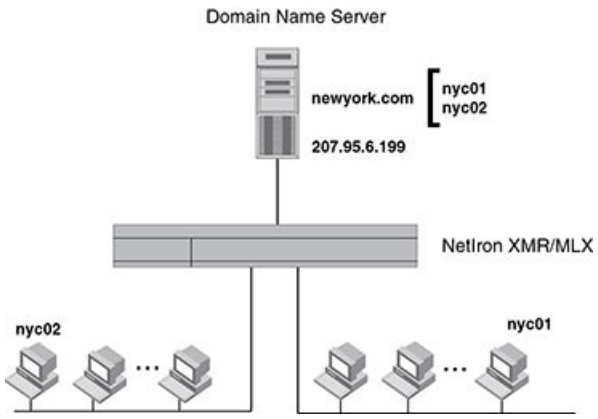
For IPv6 DNS record queries:

- Loop thru all configured IPv6 DNS servers,
- If no IPv6 DNS servers were configured, then loop through all configured IPv4 DNS servers (if any).

Using a DNS name to initiate a trace route

Suppose you want to trace the route from a Brocade device to a remote server identified as NYC02 on domain newyork.com.

FIGURE 185 Querying a host on the newyork.com domain



Because the newyork.com domain is already defined on the Brocade device, you need to enter only the host name, NYC02, as noted below.

```
device# traceroute nyc02
```

Syntax: [no] traceroute host-ip-addr [**maxttl value**] [**minttl value**] [**numeric**] [**timeout value**] [**source-ip ip addr**]

The only required parameter is the IP address of the host at the other end of the route.

After you enter the command, a message indicating that the DNS query is in process and the current gateway address (IP address of the domain name server) being queried appear on the screen.

```
Type Control-c to abort
Sending DNS Query to 10.157.22.199
Tracing Route to IP node 10.157.22.80
To ABORT Trace Route, Please use stop-traceroute command.
 Traced route to target IP node 10.157.22.80:
   IP Address          Round Trip Time1    Round Trip Time2
  10.95.6.30         93 msec             121 msec
```

NOTE
In the above example, 10.157.22.199 is the IP address of the domain name server (default DNS gateway address), and 10.157.22.80 represents the IP address of the NYC02 host.

Using Telnet and Secure Shell

Up to six inbound and five outbound Telnet connections can be supported simultaneously by the Brocade device. The Brocade device also supports Secure Shell (SSH) access to management functions.

Changing the encapsulation type for IP packets

The Brocade device encapsulates IP packets into Layer 2 packets, to send the IP packets on the network. A Layer 2 packet is also called a MAC layer packet or an Ethernet frame. The MAC address of the Brocade device interface sending the packet is the source address of the Layer 2 packet. The Layer 2 packet's destination address can be one of the following:

- The MAC address of the IP packet's destination. In this case, the destination device is directly connected to the Brocade device.
- The MAC address of the next-hop gateway toward the packet's destination.
- An Ethernet broadcast address.

The entire IP packet, including the source address, destination address, other control information, and the data, is placed in the data portion of the Layer 2 packet. Typically, an Ethernet network uses one of two different formats of Layer 2 packet:

- Ethernet II
- Ethernet SNAP (also called IEEE 802.3)

The control portions of these packets differ slightly. All IP devices on an Ethernet network must use the same format. The Brocade device uses Ethernet II by default. You can change the IP encapsulation to Ethernet SNAP on individual ports if needed.

NOTE
All devices connected to the Brocade device port must use the same encapsulation type.

To change the IP encapsulation type on interface 1/5 to Ethernet SNAP, enter the following commands.

```
device(config)# int e 1/5
device(config-if-e1000-1/5)# ip encapsulation snap
```

Syntax: [no] ip encapsulation snap | ethernet-2

Setting the maximum frame size globally

You can set the default maximum frame size to control the maximum size of Ethernet frames that the Ethernet MAC framers will accept or transmit. The size is counted from the beginning of Ethernet header to the end of CRC field. The default maximum frame size must be greater than an IP MTU value set using the Globally changing the IP MTU on page 708.

To set a maximum frame size that applies to the device for Ethernet ports, enter a command such as the following.

```
device(config)# default-max-frame-size 2000
device(config)# write memory
device(config)# reload
```

Syntax: [no] default-max-frame-size bytes

Enter 1298 - 9216 for *bytes*. On Brocade NetIron XMR Series and Brocade NetIron MLX Series devices, the default is 1548 bytes for Ethernet ports.

On Brocade NetIron CES Series devices, the *bytes* variable specifies a even number of bytes between 1298 - 9216. The default value is 1548 bytes.

NOTE
You must run the **write memory** command and reload the Brocade device for the **default-max-frame-size** command to take effect.

NOTE
In a VLAN-tagged port, the device can accept a frame size up to the default maximum frame size with or without the VLAN-tagged frame. However, it can only transmit a frame size up to the default maximum frame size plus vlan tag 4 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 Brocade device fragments the IP packet into multiple parts that will fit into Layer 2 packets, and sends the parts of the fragmented IP packet separately, in different Layer 2 packets. The device that receives the multiple fragments of the IP packet reassembles the fragments into the original packet. The default IP MTU is 1500 bytes for Ethernet II packets. You can change the IP MTU globally or for individual IP interfaces. You can increase the IP MTU size to accommodate large packet sizes, such as jumbo packets, globally or on individual IP interfaces. However, IP MTU cannot be set higher than the maximum frame size, minus 18.

NOTE
For multicast data traffic, frames are not fragmented and the IP MTU setting is ignored.

For jumbo packets, the Brocade device supports hardware forwarding of Layer 3 jumbo packets. Layer 3 IP unicast jumbo packets received on a port that supports the frame's IP MTU size and forwarded to another port that also supports the frame's IP MTU size are forwarded in hardware.

NOTE
Policy Based Routing (PBR) currently does not support this IP MTU feature.

Configuration considerations for increasing the IP MTU:

- The maximum value of an IP MTU cannot exceed the configured maximum frame size, minus 18. For example, global IP MTU cannot exceed the value of **default-max-frame-size** , minus 18 bytes. IP MTU for an interface cannot exceed the value of the maximum frame size configured, minus 18 bytes. The 18 bytes are used for Ethernet header and CRC.
- When you increase the IP MTU size of for an IP interface, the increase uses system resources. Increase the IP MTU size only on the IP interfaces that need it. For example, if you have one IP interface connected to a server that uses jumbo frames and two other IP interfaces connected to clients that can support the jumbo frames, increase the IP MTU only on those three IP interfaces. Leave the IP MTU size on the other IP interfaces at the default value (1500 bytes). Globally increase the IP MTU size only if needed.
- The difference between IP MTU and **default-max-frame size** should be as follows.
  - 18 bytes for untagged packets
  - 22 bytes for single-tagged packets and
  - 26 bytes for dual-tagged packets

How To determine the actual MTU value

An IPv4 interface can obtain it's MTU value from any of the following sources:

- Default IP MTU setting
- Global MTU Setting

- Interface MTU Setting

An interface determines its actual MTU value through the process described below.

1. If an IPv4 Interface MTU value is configured, that value will be used.
2. If an IPv4 Interface MTU value is not configured and an IPv4 Global MTU value is configured, the configured global MTU value will be used.
3. If neither an IPv4 Interface MTU value or an IPv4 Global MTU value are configured, the default IPv4 MTU value of 1500 will be used.

Globally changing the IP MTU

To globally enable jumbo support on all IP interfaces, enter commands such as the following.

```
device(config)# ip global-mtu 5000
device(config)# write memory
```

Syntax: [no] ip global-mtu bytes

The *bytes* parameter specifies the maximum IP packet size to be forwarded on a port. You may enter any number within the range of 576 - 9198. However, this value must be 18 bytes less than the value of the global maximum frame size.

NOTE
The global IP MTU change does not get applied to IP tunnel interfaces such as GRE interface. The MTU for these interfaces has to be changed on interface level.

Changing the maximum transmission unit on an individual interface

By default, the maximum IP MTU sizes are as follows:

- 1500 bytes - The maximum for Ethernet II encapsulation

NOTE
The IP MTU configured at the IP interface level takes precedence over the IP MTU configured at the global level for that IP interface.

To change the IP MTU for interface 1/5 to 1000, enter the following commands.

```
device(config)# int e 1/5
device(config-if-e10000-5)# ip mtu 1000
```

Syntax: [no] ip mtu bytes

The *bytes* variable specifies the IP MTU. However, the value of IP MTU on an interface cannot exceed the configured value **default-max-frame-size** , minus 18 bytes. The default IP MTU for Ethernet II packets is 1500.

If the interface is part of a VLAN, then ensure that you change the IP MTU only at the VE interface and not at the physical port. To change the IP MTU at the VE interface, enter the following commands:

```
Brocade(config)# int ve 103
Brocade(config-vif-103)# ip mtu 1000
```

NOTE
All member ports of a VLAN will have the same IP MTU value as the VE interface.

Changing the router ID

In most configurations, a Brocade device has multiple IP addresses, usually configured on different interfaces. As a result, a Brocade device's identity to other devices varies depending on the interface to which the other device is attached. Some routing protocols, including OSPF and BGP4, identify a Brocade device by just one of the IP addresses configured on the Brocade device, regardless of the interfaces that connect the Brocade 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 Brocade device is one of the following:

- If the device has loopback interfaces, the default router ID is the IP address configured on the lowest numbered loopback interface configured on the Brocade device. For example, if you configure loopback interfaces 1, 2, and 3 as follows, the default router ID is 10.9.9.9/24:
  - Loopback interface 1, 10.9.9.9/24
  - Loopback interface 2, 10.4.4.4/24
  - Loopback interface 3, 10.1.1.1/24
- If the IP address from loopback1 interface (lowest numbered loopback interface) is removed, the next lowest loopback interface IP address is selected as router-id.
- If a loopback interface is not configured, then the lowest IP address configured over the physical interface is selected as the router ID.

If you prefer, you can explicitly set the router ID to any valid IP address. The IP address should not be in use on another device in the network.

You can set a router ID for a specific VRF as described within this section. In order to make the route ID calculation more deterministic, the device calculates the router-id value during bootup and does not calculate or change the router-id value unless the IP address used for the router-id value on the device is deleted, or the **clear router-id** command is issued. Additionally, setting a router-id value overrides the existing router-id value and takes effect immediately. Once a router-id value set by a user is removed using the **no ip router-id** command, the device will again recalculate the router-id value based on current information.

NOTE
The Brocade device uses the same router ID for both OSPF and BGP4. If the device is already configured for OSPF, you may want to use the router ID that is already in use on the device rather than set a new one. To display the router ID, enter the **show ip** command at any CLI level.

To change the router ID, enter a command such as the following.

```
device(config)# ip router-id 10.157.22.26
```

Syntax: [no] ip router-id ip-addr

The *ip-addr* can be any valid, unique IP address.

To set the router ID within a VRF, enter a command such as the following.

```
device(config)# vrf blue
device(config-vrf-blue)# ip router-id 10.157.22.26
```

Syntax: [no] ip router-id ip-addr

NOTE
The command for setting the router ID for a specified VRF is exactly the same as for the default VRF. The only difference is that when setting it for a specific VRF, the **ip router-id** command is configured within the VRF as shown in the example.

NOTE
You can specify an IP address used for an interface, but do not specify an IP address in use by another device.

Recalculating the router ID

You can use the **clear ip router-id** command to direct a device to recalculate the IP router ID. This can be done for the default VRF or for a specified VRF, as shown in the following.

```
device(config)# clear ip router-id
```

Syntax: clear ip router-id [**vrf** *vrf-name*]

Using this command without the **vrf** option recalculates the IP router ID for the default VRF.

You can use the **vrf** option to recalculate the IP router ID for a specific VRF that is specified by the *vrf-name* variable.

IPv6 ND Global Router Advertisement Control

IPv6 ND Global Router Advertisement Control allows for disabling sending out router advertisements at the global system level. The **no ipv6 nd global-suppress-ra** command at the interface level allows the user to disable and enable the sending of the ND Router Advertisement on an interface. By default, the sending of ND Router Advertisement (RA) is enabled on all interfaces, except for the tunnel and loopback interfaces, providing that the IPv6 Unicast Routing is enabled and the interfaces are active for IPv6.

The IPv6 ND Global Router Advertisement Control gives the ability to quickly turn off the sending of IPv6 ND Router Advertisement message on all IPv6 enabled interfaces.

By default,

- The ND Router Advertisement is enabled.
- Interface is enabled to send ND Router Advertisements.
- The **ipv6 nd suppress-ra** and **ipv6 nd send-ra** interface commands, when configured, override the system and VRF global **ipv6 nd global-suppress-ra** command.

Users sometimes require the ability to quickly turn off the sending of IPv6 ND Router Advertisement message on all IPv6 enabled interfaces. This is achieved by providing the following additional configuration command at system and VRF level:

```
device(config-vrf-red-ipv6)[no]ipv6 nd global-suppress-ra
```

The **ipv6 nd send-ra** command is a new interface level command added as part of this enhancement. This allows the user to configure the sending of RA messages on some selected interfaces when the **ipv6 nd global-suppress-ra** command is set to disable the sending of RA messages on all other interfaces.

Syntax: [no] ipv6 nd global-suppress-ra

Configuring IPv6 ND global router advertisement globally on the default VRF

When configuring the **ipv6 nd global-suppress-ra** command, the ND Router Advertisement messages is not sent out on any interface in the default VRF, unless the **ipv6 nd send-ra** is set on the interface. By default, **ipv6 nd global-suppress-ra** is not set for the IPv6 VRF.

Use the following command under **address-family ipv6** for a specific VRF is added and applies to the IPv6 VRF:

```
device(config)# vrf red
device(config-vrf-red)#address-family ipv6
device(config-vrf-red-ipv6)#ipv6 nd global-suppress-ra
```

Syntax: [no] ipv6 nd global-suppress-ra

The following command when set ensures that IPv6 ND Router Advertisement messages are sent out on the interface regardless of the setting of the **ipv6 nd global-suppress-ra** for the interface's VRF.

Syntax: [no] ipv6 nd send-ra

By default, **ipv6 nd send-ra** is not set on the interface. When **ipv6 nd send-ra** is set, the **ipv6 nd suppress-ra** command is unset. However, **ipv6 nd suppress-ra** is not set when **ipv6 nd send-ra** is issued on the interface. This is similar to when a user issue existing **ipv6 nd suppress-ra** command is on an interface, the **ipv6 nd send-ra** is unset. By default, **ipv6 nd suppress-ra** is not set.

If sending of RA messages is required on some selected interfaces to continue, then you must set the **ipv6 nd send-ra** command on these interfaces before setting the **ipv6 nd global-suppress-ra** command to disable the sending of RA messages on all other interfaces. Otherwise, the RA messages are not sent out until the **ipv6 nd send-ra** command is set on each of the selected interfaces.

The interface **ipv6 nd send-ra** and **ipv6 nd suppress-ra** commands are sticky in that they are independent of the **ipv6 nd global-suppress-ra** command and **either ipv6 nd send-ra** or **ipv6 nd suppress-ra** can still be present in configuration even when the **ipv6 nd global-suppress-ra** is also in configuration.

Show commands

The output of **show ipv6 interface** command is modified when the sending of router advertisement is disabled on the interface or globally. Use the **show ipv6 interface** command to display the output of the interface.

```
device#show ipv6 int eth 2/1
Interface Ethernet 2/1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::200:ff:fe03:c030 [Preferred]
  Global unicast address(es):
    31:1:1::3 [Preferred],  subnet is 31:1:1::/64
    31:1:1:: [Anycast],  subnet is 31:1:1::/64
  Joined group address(es):
    ff02::6
    ff02::5
    ff02::1:ff00:3
    ff02::1:ff03:c030
    ff02::2
    ff02::1
  Port belongs to VRF: default-vrf
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1000 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements suppressed
    No Inbound Access List Set
  No Outbound Access List Set
  IPv6 RPF mode: None  IPv6 RPF Log: Disabled
  OSPF enabled
  RxPkts:       0                       TxPkts:   0
  RxBytes:      0                       TxBytes:  0
  IPv6 unicast RPF drop: 0
  IPv6 unicast RPF suppressed drop: 0
device#
```

Syntax: show ipv6 interface [*interface* [*port-number* | *number*]]

The *interface* parameter displays detailed information for a specified interface. For the interface, the user can specify the Ethernet, loopback, tunnel, or VE keywords. If the user specifies an Ethernet interface, then the user must also specify the port number associated with the interface. If the user specifies a loopback, tunnel, or VE interface, the user must also specify the number associated with the interface.

Table 82 defines the **show ipv6 interface** command output display that shows the following information:

TABLE 82 General IPv6 interface information fields

This field...	Displays...
Routing protocols	A one-letter code that represents a routing protocol that can be enabled on an interface.
Interface	The interface type, and the port number or number of the interface.
Status	The status of the interface. The entry in the Status field will be either "up/up" or "down/down".
Routing	The routing protocols enabled on the interface.
Global Unicast Address	The global unicast address of the interface.

Specifying a single source interface for Telnet, SSH, NTP, TFTP, TACACS/TACACS+, or RADIUS packets

When the Brocade device originates a Telnet, SSH, NTP, TFTP, TACACS/TACACS+, or RADIUS packet, the source address of the packet is the lowest-numbered IP address on the interface that sends the packet. You can configure the Brocade device to always use the lowest-numbered IP address on a specific interface as the source addresses for these types of packets. When you configure the Brocade device to use a single source interface for all Telnet, TACACS/TACACS+, or RADIUS packets, the Brocade device uses the same IP address as the source for all packets of the specified type, regardless of the ports that actually sends the packets.

Identifying a single source IP address for Telnet, SSH, NTP, TFTP, TACACS/TACACS+, or RADIUS packets provides the following benefits:

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

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

Configuring an interface as the source for Syslog packets

You can configure the device to use the lowest-numbered IP or IPv6 address configured on a loopback interface, virtual interface, or Ethernet port as the source for all Syslog packets from the device. The software uses the lowest-numbered IP or IPv6 address configured on the interface as the source IP address for the packets.

For example, to specify the lowest-numbered IP address configured on a virtual interface as the device's source for all Syslog packets, enter commands such as the following.

```
device(config)# int ve 1
device(config-vif-1)# ip address 10.0.0.4/24
device(config-vif-1)# exit
device(config)# ip syslog source-interface ve 1
```

The commands in this example configure virtual interface 1, assign IP address 10.0.0.4/24 to the interface, then designate the interface's address as the source address for all Syslog packets.

Syntax: [no] ip syslog source-interface ethernet [**slotnum/**] **portnum** | **loopback num** | **ve num**

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

The default is the lowest-numbered IP or IPv6 address configured on the port through which the packet is sent. The address therefore changes, by default, depending on the port.

With this new command, the source ip of syslog is no longer controlled by the **snmp-server trap-source** command.

Configuring ARP parameters

Address Resolution Protocol (ARP) is a standard IP protocol that enables the Brocade device to obtain the MAC address of another device's interface when the Brocade device knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

How ARP works

The Brocade device needs to know a destination's MAC address when forwarding traffic, because the Brocade device encapsulates the IP packet in a Layer 2 packet (MAC layer packet) and sends the Layer 2 packet to a MAC interface on a device directly attached to the Brocade device. The device can be the packet's final destination or the next-hop router toward the destination.

The Brocade device encapsulates IP packets in Layer 2 packets regardless of whether the ultimate destination is locally attached or is multiple router hops away. Since the Brocade device's IP route table and IP forwarding cache contain IP address information but not MAC address information, the Brocade device cannot forward IP packets based solely on the information in the route table or forwarding cache. The Brocade device needs to know the MAC address that corresponds with the IP address of either the packet's locally attached destination or the next-hop router that leads to the destination.

For example, to forward a packet whose destination is multiple router hops away, the Brocade device must send the packet to the next-hop router toward its destination, or to a default route or default network route if the IP route table does not contain a route to the packet's destination. In each case, the Brocade device must encapsulate the packet and address it to the MAC address of a locally attached device, the next-hop router toward the IP packet's destination.

To obtain the MAC address required for forwarding a datagram, the Brocade device does the following:

- First, the Brocade device looks in the ARP cache (not the static ARP table) for an entry that lists the MAC address for the IP address. The ARP cache maps IP addresses to MAC addresses. The cache also lists the port attached to the device and, if the entry is dynamic, the age of the entry. A dynamic ARP entry enters the cache when the Brocade device receives an ARP reply or receives an ARP request (which contains the sender's IP address and MAC address). A static entry enters the ARP cache from the static ARP table (which is a separate table) when the interface for the entry comes up. To ensure the accuracy of the ARP cache, each dynamic entry has its own age timer. The timer is reset to zero each time the Brocade device receives an ARP reply or ARP request containing the IP address and MAC address of the entry. If a dynamic entry reaches its maximum allowable age, the entry times out and the software removes the entry from the table. Static entries do not age out and can be removed only by you.

- If the ARP cache does not contain an entry for the destination IP address, the Brocade device broadcasts an ARP request out all its IP interfaces. The ARP request contains the IP address of the destination. If the device with the IP address is directly attached to the Brocade device, the device sends an ARP response containing its MAC address. The response is a unicast packet addressed directly to the Brocade device. The Brocade device places the information from the ARP response into the ARP cache. ARP requests contain the IP address and MAC address of the sender, so all devices that receive the request learn the MAC address and IP address of the sender and can update their own ARP caches accordingly.

  NOTE
  The ARP request broadcast is a MAC broadcast, which means the broadcast goes only to devices that are directly attached to the Brocade device. A MAC broadcast is not routed to other networks. However, some routers, including the Brocade device, can be configured to reply to ARP requests from one network on behalf of devices on another network. Refer to Enabling proxy ARP on page 715.

  NOTE
  If the device 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 Brocade device knows of no route to the destination address), the device sends an ICMP Host Unreachable message to the source.

Rate limiting ARP packets

For rate-limiting purposes, ARP traffic destined for the CPU is assigned a separate global QoS ID 0xFFE. You can configure the rate-limit parameters using the following global CONFIG command.

```
device(config)# ip rate-limit arp policy-map policy-map-name
```

By default, the rate-limit parameters for QoS ID 0xFFE will be initialized to allow line-rate traffic. The rate-limit parameters specified using the policy-map are applicable on a per-PPCR basis.

To display ARP accounting statistics, enter the following command.

```
device(config)# show rate-limit arp
```

This command displays the byte counters corresponding to QoS ID 0xFFE.

```
device(config)# clear rate-limit arp
```

This command clears the byte counters corresponding to QoS ID 0xFFE.

When priority-based rate limiting is enabled, QoS IDs 0x3FE, 0x7FE and 0xBFE will be re-mapped to 0xFFE. When priority-based rate limiting is disabled, QoS IDs 0x3FE, 0x7FE and 0xBFE will not be re-mapped to 0xFFE. In either case, only QoS ID 0xFFE will be added to the list of used QoS IDs.

To enable the dynamic addition, deletion, or change in rate-limit values of a policy-map, enter the following command.

```
device(config)# ip rate-limit arp policy-map policy-map-name
```

This command takes effect automatically, without unbinding and rebinding the ARP RL policy. If the ARP Rate Limit policy specifies an undefined policy-map, rate limit values are initialized to line-rate values. Dynamic enabling and disabling of priority based rate limiting on a global basis takes effect automatically for the ARP RL policy.

NOTE
ARP packets destined for the CPU will be not be rate-limited by interface-level Layer 2 RL-ACLs. To rate-limit switched ARP packets using interface-level Layer 2 ACLs, you must define an explicit ACL filter with an "etype arp" option, as shown in the following example:

To define an explicit ACL filter, enter commands similar to the following.

```
device(config)# access-list 410 permit any any any etype arp
device(config)# int eth 4/1
device(config-if-e10000-4/1)# rate-limit in access-gr 410 policy-map view
```

NOTE
Since ARP packets are broadcast packets, ARP packets are switched by default within a VLAN by the CPU. Thus to rate-limit switched ARP packets using interface-level Layer 2 ACLs, you must also configure vlan-cpu-protection.

Changing the ARP aging period

When the Brocade device places an entry in the ARP cache, the Brocade device also starts an aging timer for the entry. The aging timer ensures that the ARP cache does not retain learned entries that are no longer valid. An entry can become invalid when the device with the MAC address of the entry is no longer on the network. The underlying MAC aging out causes deletion of the corresponding ARP entries.

The ARP age affects dynamic (learned) entries only, not static entries. The default ARP age is ten minutes. On the Brocade device, you can change the ARP age to a value from 0 through 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.

```
device(config)# ip arp-age 20
```

Syntax: [no] ip arp-age num

The *num* parameter specifies the number of minutes and can be from 0 through 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.

```
device(config-if-e1000-1/1)# ip arp-age 30
```

Enabling proxy ARP

Proxy ARP allows the Brocade device to answer ARP requests from devices on one network on behalf of devices in another network. Since ARP requests are MAC-layer broadcasts, they reach only the devices that are directly connected to the sender of the ARP request. Thus, ARP requests do not cross routers.

For example, if Proxy ARP is enabled on the Brocade device connected to two subnets, 10.10.10.0/24 and 10.20.20.0/24, the Brocade device can respond to an ARP request from 10.10.10.69 for the MAC address of the device with IP address 10.20.20.69. In standard ARP, a request from a device in the 10.10.10.0/24 subnet cannot reach a device in the 10.20.20.0 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.

```
device(config)# ip proxy-arp
```

To again disable IP proxy ARP, enter the following command.

```
device(config)# no ip proxy-arp
```

Syntax: [no] ip proxy-arp

Enabling local proxy ARP

Under some Layer-2 configurations such as uplink-switch or private VLAN, broadcast packets are not flooded to every port in a VLAN. In these configurations, an ARP request from one host may not reach another host. Enabling the Local Proxy ARP feature on a port directs the device to reply on behalf of a target host if it exists. The ARP reply returned contains the device's mac address instead of the mac address of the target host. In this transaction, the traffic sent to the target host is Layer-3 forwarded rather than Layer-2 switched.

To enable Local Proxy ARP, the global-level command **ip proxy-arp** must first be enabled as described in Enabling proxy ARP on page 715. After **ip proxy-arp** has been enabled globally, Local Proxy ARP can be enabled on a specified interface using the following command.

```
device(config-if-e1000-1/1)# interface ethernet 1/1
device(config-if-e1000-1/1)# ip local-proxy-arp
```

Syntax: [no] ip local-proxy-arp

Disabling gratuitous ARP requests for local proxy ARP

When the Local Proxy ARP is configured under the IP interface, the Brocade device will reply to ARP requests on behalf of the hosts inside the subnet using its own MAC address. Refer to Enabling local proxy ARP on page 716 for information on configuring the **Local Proxy ARP** command. In this configuration, when a host comes up, the host tries to ping its own IP address to make sure there is no duplicated IP address by issuing a gratuitous ARP request to its only IP address. The Brocade device will reply to this request because it is required under the Local Proxy ARP configuration. When the host receives the ARP reply, the host incorrectly assumes that there is another host using the same IP address.

A gratuitous ARP request packet is defined as an ARP request packet with the sender protocol address that equals to the target protocol address. By disabling Gratuitous ARP Requests for Local Proxy ARP, you are able to control whether to reply to gratuitous ARP requests under the Local Proxy ARP configuration.

To enable the ignore-gratuitous-arp parameter when the **ip local-proxy-arp** command is turned on, enter the following command.

```
device(config-if-e1000-1/6)# ip local-proxy-arp ignore-gratuitous-arp
```

To disable only the ignore-gratuitous-arp parameter when the Local Proxy ARP is configured, enter the following command.

```
device(config-if-e1000-1/6)# no ip local-proxy-arp ignore-gratuitous-arp
```

To disable both the **Local Proxy ARP** command and the ignore-gratuitous-arp parameter, enter the following command.

```
device(config-if-e1000-1/6)# no ip-local-proxy-arp
```

Syntax: [no] ip local-proxy-arp [**ignore-gratuitous-arp**]

When using the **no ip local-proxy-arp ignore-gratuitous-arp** command, only the ignore-gratuitous-arp parameter is turned off. The **ip-local-proxy-arp** command is still turned on.

The Brocade device drops all ARP packets that are sent from its own interface. When the ignore-gratuitous-arp parameter is turned on, the Brocade device will not reply to a gratuitous ARP request even if the target protocol address matches the configured interface IP address.

Creating static ARP entries

The Brocade device has a static ARP table, in addition to the regular ARP cache. The static ARP table contains entries that you configure.

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the Brocade device, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the device receives an ARP request from the device that has the entry's address.

You can increase the number of configurable static ARP entries. Refer to Changing the ARP timer on page 717.

To display the ARP cache and static ARP table, refer to the following:

- To display the ARP table, refer to Displaying the ARP cache on page 793.
- To display the static ARP table, refer to Displaying the static ARP table on page 794.
- To create a static ARP entry for a static MAC entry, refer to Creating ARP entries on page 721.

Changing the ARP timer

When an entry is initially added to the ARP table, it is listed as "Pending." When it is in this state, a series of ARP requests are made to determine if it is a valid entry. If the first attempt succeeds, the status of the entry is changed to "dynamic." It is then subject to the normal rules for dynamic entries. If three attempts fail, the entry is removed from the table.

The ARP timer determines the amount of time that elapses after the ARP request is sent before determining that the request has failed.The **arp-timer** command allows you change the length of the ARP timer as shown in the following.

```
device(config)# ip arp-timer 12
```

Syntax: [no] ip arp-timer timer-value

The *timer-value* variable has now been changed so that you are able to enter a value between 1 and 500. Each increment represents 100 ms. Consequently, the minimum value of 1 equals 100 ms.

The default value is 10 which equals 1 sec.

This value can be used to adjust how frequently an ARP request is sent out for a pending ARP entry.

Changing the ARP pending retry timer

The ARP Pending Retry Timer for Brocade device will send out three ARP request packets for the configured period until ARP is resolved to prevent large amounts of ARP requests from flooding the network during network host scanning activity. The ARP Pending Retry Timer is configurable depending upon the requirements of your system configurations.

The **arp-pending-retry-timer** command allows you to change the length of the ARP pending retry timer as shown in the following.

```
device(config)# ip arp-pending-retry-timer 120
```

Syntax: [no] ip arp-pending-retry-timer timer-value

The *timer-value* variable is a value between 10 to 3600 seconds. The default value is 60 seconds.

Generating syslog notification for differing Ethernet source MAC and ARP sender MAC addresses

The Brocade devices generate a syslog notification whenever there is a mismatch between the Layer 2 header source MAC address and the ARP sender MAC address.

This syslog notification is supported on the Brocade NetIron XMR Series, Brocade NetIron MLX Series, Brocade NetIron CER Series, and the Brocade NetIron CES Series platforms.

Configuration step

Enter the **logging enable mac-mismatch-detection** command for syslog notification due to MAC address mismatch.

The syslog message helps you identify the root cause for the traffic outage scenario and you can proceed with the static MAC address workaround in MCT by configuring the static MAC address in the CCEP port. The following syslog message is displayed when there is a MAC address mismatch.

```
SYSLOG: <14>Dec 16 05:53:23 MLX_1 MAC_MISMATCH_DETECTION: ARP pkt received with diff eth source MAC
and diff ARP sender MAC. Eth src MAC: 0024.3892.4c02 ARP sender MAC: 0034.2867.2c01.
```

Dynamic ARP inspection

NOTE
This feature is supported on Layer 2 and Layer 3 code.

Dynamic ARP Inspection (DAI) enables the device to intercept and examine all ARP request and response packets in a subnet and discard those packets with invalid IP to MAC address bindings. DAI can prevent common man-in-the-middle (MiM) attacks such as ARP cache poisoning, and disallow mis-configuration of client IP addresses.

ARP poisoning

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. Before a host can talk to another host, it must map the IP address to a MAC address first. If the host does not have the mapping in its DAI table, it creates an ARP request to resolve the mapping. All computers on the subnet will receive and process the ARP requests, and the host whose IP address matches the IP address in the request will send an ARP reply.

An ARP poisoning attack can target hosts, switches, and routers connected to the Layer 2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting traffic intended for other hosts on the subnet. For instance, a malicious host can reply to an ARP request with its own MAC address, thereby causing other hosts on the same subnet to store this information in their DAI tables or replace the existing ARP entry. Furthermore, a host can send gratuitous replies without having received any ARP requests. A malicious host can also send out ARP packets claiming to have an IP address that actually belongs to another host (e.g. the default router). After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

How DAI works

DAI allows only valid ARP requests and responses to be forwarded.

A device on which ARP Inspection is configured does the following:

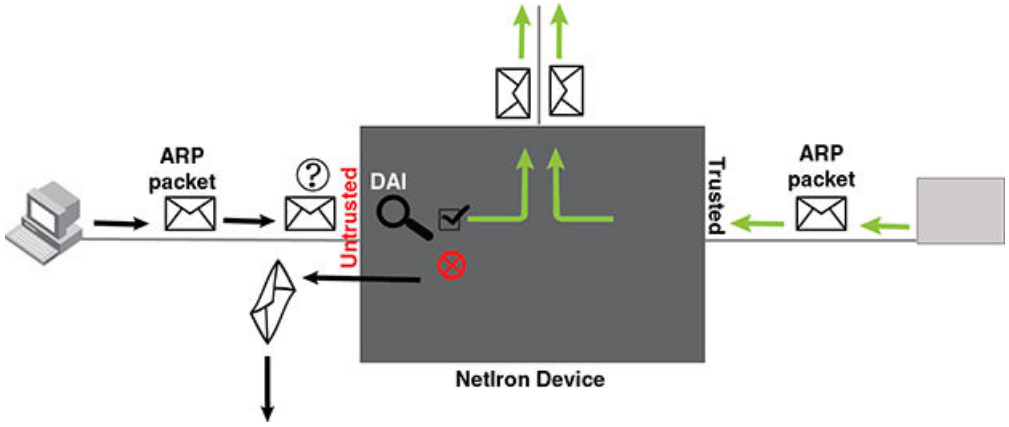
- Intercepts ARP packets received by the system CPU
- Inspects all ARP requests and responses received on untrusted ports
- Verifies that each of the intercepted packets has a valid IP-to-MAC address binding before updating the ARP table, or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

When you enable ARP Inspection on a VLAN, by default, all member ports are untrusted. You must manually configure trusted ports. In a typical network configuration, ports connected to host ports are untrusted. You configure ports connected to other switches or routers as trusted.

DAI inspects ARP packets received on untrusted ports, as shown in Figure 189. DAI carries out the inspection based on IP-to-MAC address bindings stored in a trusted binding database. For the Brocade device, the binding database is the ARP table, which supports DAI, DHCP snooping, and IP Source Guard. To inspect an ARP request packet, DAI checks the source IP and source MAC address against the ARP table. For an ARP reply packet, DAI checks the source IP, source MAC, destination IP, and destination MAC addresses. DAI forwards the valid packets and discards those with invalid IP-to-MAC address bindings.

When ARP packets reach a trusted port, DAI lets them through, as shown in Figure 189.

FIGURE 186 Dynamic ARP inspection at work



ARP entries

DAI uses the IP/MAC mappings in the ARP table to validate ARP packets received on untrusted ports. ARP entries in the ARP table derive from the following:

- **ARP Inspection** - statically configured VRF+VLAN +IP/MAC mapping.
- **ARP** - statically configured VRF+IP/MAC/port mapping.
- **DHCP-Snooping ARP** - information collected from snooping DHCP packets when DHCP snooping is enabled on VLANs.

Configuring DAI

NOTE
An index number is no longer needed to configure static ARP entries.

Follow the steps listed below to configure DAI.

1. Configure inspection of ARP entries for hosts on untrusted ports. Enable ARP Inspection on a VLAN to inspect ARP packets.
2. Configure the trust settings of the VLAN members. ARP packets received on *trusted* ports bypass the DAI validation process. ARP packets received on untrusted ports go through the DAI validation process.
3. Enable DHCP snooping to populate the DHCP snooping IP-to-MAC binding database. **Refer to** DHCP binding database on page 727 **for more information.**

   The following shows the default settings of ARP Inspection.

Feature	Default
Dynamic ARP Inspection	Disabled
Trust setting for ports	Untrusted

Enabling dynamic ARP inspection on a VLAN

ARP and Dynamic inspection ARP entries need to be configured for hosts on untrusted ports. Otherwise, when Dynamic ARP Inspection checks ARP packets from these hosts against entries in the ARP table, it will not find any entries for them, and the device will not allow and learn ARP from an untrusted host.

Dynamic ARP Inspection is disabled by default. To enable Dynamic ARP Inspection on an existing VLAN or a range of VLANs, enter the following command.

```
device(config)# ip arp-inspection vlan 18 to 20
```

The command enables Dynamic ARP Inspection on VLAN 18 through VLAN 20. ARP packets from untrusted ports in VLAN 18 through VLAN 20 will undergo Dynamic ARP Inspection.

Syntax: [no] ip-arp inspection vlan vlan_id to vlan_id

The *vlan_id* variable specifies the ID of a configured VLAN or VLAN range. Valid VLAN ranges are 1-4090.

Configuring static ARP on a VLAN and port

In the Brocade device configuration, the DHCP binding database is integrated with the ARP Inspection table. The ARP inspection table stores the DAI IP/MAC binding information, which is used to build the IP source guard ACL. The **static arp** command allows you to configure both the vlan id and port parameters on a layer 2 interface.

To configure a static arp entry for a vlan id, enter the following command.

```
device(config)#arp 10.1.0.2 aabb.cc00.0100 vlan 10
```

Syntax: [no] arp ip mac [**vlan vlan_id**] [**port**]

The *ip* variable specifies the IP address for the static IP ARP entry.

The *mac* variable specifies the MAC address for the static IP ARP entry.

The *vlan_id* variable configures the static ARP entry for a vlan. The VLAN ID range is 1-4090.

The *port* variable configures the static ARP entry for a port.

If the vlan id is not configured when IP source guard is turned on, the IP address is assumed to be valid on all the vlans on the port.

If both the vlan id and the port are not configured when IP source guard is turned on, the IP address is assumed to be valid for all vlans.

Enabling trust on a port

The default trust setting for a port is untrusted. For ports that are connected to host ports, leave their trust settings as untrusted.

To enable trust on a port, enter commands such as the following.

```
device(config)# interface ethernet 1/4
device(config-if-e10000-1/4)# arp-inspection-trust
```

The commands change the CLI to the interface configuration level of port 1/4 and set the trust setting of port 1/4 to trusted.

Syntax: [no] arp-inspection-trust

Creating ARP entries

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the Brocade device, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the device receives an ARP request from the device that has the entry's address.

To create a static ARP entry for a static MAC entry, enter a command such as the following.

```
device(config)# arp 10.53.4.2 1245.7654.2348 vlan 10
```

The command adds a static ARP entry that maps IP address 10.53.4.2 to MAC address 1245.7654.2348. The entry is for a MAC address connected to VLAN 10 of the Brocade device.

Syntax: [no] arp ip-addr mac-addr [**ethernet slot/port** | **vlan vlan_id**]

The *ip-addr* parameter specifies the IP address of the device that has the MAC address of the entry.

The *mac-addr* parameter specifies the MAC address of the entry.

The **ethernet***slot/port* command specifies the port number attached to the device that has the MAC address of the entry.

The **vlan** *vlan_id* variable specifies the ID of a configured VLAN or VLAN range. Valid VLAN ranges are 1-4090.

Creating multi-port ARP entries

On the device devices, multiple ports belonging to the same VE can be assigned to a single static ARP.

> NOTE
> The multi-port ARP feature can be used in a pure Layer 3 forwarding environment to forward IPv4 traffic to multiple ports and should not be used in conjunction with Multi-port static MAC.

To create a multi-port static ARP entry, enter a command such as the following.

```
device(config)# arp 10.53.4.2 1245.7654.2348 multi-ports ethernet 2/1 to 2/7 ethernet 3/1 to 3/2
```

The command above adds a static ARP entry that maps IP address 10.53.4.2 to MAC address 1245.7654.2348. If all conditions are met and the multi-port static ARP entry is instantiated in the dynamic ARP table, then packets with a destination IP address of 10.53.4.2 will be sent out on Ethernet ports 2/1-2/7 and 3/1-3/2.

Syntax: [no] arp ip address mac address [**port** | **multi-ports ethernet** [**slot1/port1** | [**slot1/port1 to slot1/port**] .. **ethernet** [**slot/port to slot/port**]]

The *ip-address* parameter specifies the IP address of the device that has the MAC address of the entry.

The *mac-address* 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 **ethernet** keyword is repeated before each individual port or range of ports to be included in the multi-port ARP entry.)

VRF Considerations

The configuration command above creates a static ARP entry associated with the default VRF. To configure the multi-port ARP for a non-default VRF, first enter the configuration mode for the non-default VRF, then enter address family command mode using commands such as the following.

```
device(config)# vrf test
device(config-vrf-test)# address-family ipv4
device(config-vrf-test-ipv4)# arp 10.6.6.7 0001.0001.0001 multi-ports ethernet 2/1 to 2/7
device(config-vrf-test-ipv4)# ethernet 3/2 to 3/2
```

```
device(config-vrf-test-ipv4)# exit-address-family
device(config-vrf-test)# exit vrf
```

The above commands create a multi-port ARP entry associated with a non-default VRF called "test."

NOTE
This feature is supported on both the Brocade NetIron XMR Series, Brocade NetIron MLX Series and Brocade NetIron CER Series, Brocade NetIron CES Series series platforms.

Instantiation in the ARP table

NOTE
Configuring a multi-port static ARP entry does not automatically create a dynamic ARP entry!

NOTE
The multi-port ARP feature can be used in a pure Layer 3 forwarding environment to forward IPv4 traffic to multiple ports and should not be used in conjunction with Multi-port static MAC.

The following four conditions must be met in order for a user-created multi-port static ARP entry to be instantiated in the dynamic ARP table:

1. All the ports configured in the multi-port static ARP entry need to belong to the same VE.
2. The IP address of the multi-port static ARP entry needs to match the subnet of the VEto which the ports belong, and it must be in the same VRF.
3. At least one of the ports in the configured port list needs to be up.
4. MPLS uplink must not be configured on the VE that subnets the static ARP IP address.

   If these four conditions are met, a conflict check is performed before adding the static ARP entry to the dynamic ARP table. If a dynamic entry already exists with the same IP address and VRF, the static ARP will override the dynamic entry and packets will be forwarded to the FID for this dynamic ARP entry.

   Changes in these conditions (VE port membership changes, port up/down status changes, etc.) can trigger reevaluation of the static ARP and may result in the entry being added to or removed from the ARP table.

Supported applications

- **PBR** PBR supports use of a multi-port static ARP entry as an IP next hop.
- **Trunk ports** Primary trunk ports can be configured in multi-port static ARPs. If a secondary trunk port is included in a multi-port ARP entry, however, the trunk will not be deployed.
- **ARP inspection** ARP inspection is performed for multi-port static ARPs the same as for normal static ARP entries.

Unsupported applications

- **IP tunnel** If an IP tunnel's next hop is resolved to a multi-port static ARP entry, the tunnel will not be brought up.
- **MPLS next-hop** Configuring an MPLS uplink on the VE interface associated with a multi-port static ARP will prevent instantiation of the ARP.
- **MCT** The ICL ports in MCT and clients are not supported by multi-port static ARP and MAC.
- **PB/PBB** The non-default port types are not supported by multi-port static ARP and MAC on PB/PBB.

Creating a floating static ARP entry

You can create a static ARP entry without port assignments.

When a floating static ARP entry (Static ARP Inspection entry without port defined) is added to ARP Inspection table, the mapping is checked against the current static ARP table. If ARP entry with a matching IP but mismatch MAC is found, it will be deleted and a re-arp on the IP will be issued.

When an ARP entry is deleted from ARP Inspection table, the corresponding entry in the static ARP table will also be deleted.

To create a floating static ARP entry for a static MAC entry, enter a command such as the following.

```
device(config)# arp 10.53.4.2 1245.7654.2348
```

The command adds a floating static ARP entry that maps IP address 10.53.4.2 to MAC address 1245.7654.2348.

Syntax: [no] arp ip-addr mac-addr [**ethernet portnum** | **vlan vlan_id**]

The *ip-addr* parameter specifies the IP address of the device that has the MAC address of the entry.

The *mac-addr* parameter specifies the MAC address of the entry.

The **ethernet** *portnum* parameter specifies the port number attached to the device that has the MAC address of the entry, and is only valid for original static ARP entries.

The **vlan** *vlan_id* parameter specifies the ID of a configured VLAN.

Configuring a Virtual Routing Instance (VRF)

To configure a virtual routing instance (VRF), enter a command such as the following.

```
device(config)# vrf vpn1
```

Syntax: [no] vrf vrf-name

The **vrf** parameter specifies the virtual routing instance (VRF) specified by the variable *vrf-name*.

Adding an ARP entry for a VRF

IP Addresses can be uniquely determined by VRF. The VLAN number is not needed because the VLAN information is obtained through the ARP protocol. To define an ARP inspection entry for a specific VRF, enter commands such as the following.

```
device(config)# vrf vpn1
device(config-vrf-vpn1))#arp 10.53.4.2 1245.7654.2348 e 3/5
```

This command creates an ARP entry for vrf with IP address 10.53.4.2 and MAC address of 1245.7654.2348 on ethernet 3/5.

Syntax: [no] arp ip-addr mac-addr [**ethernet slot/port**]

The *vrf-name* parameter specifies the VRF you are configuring a static ARP entry for.

The *ip-addr* parameter specifies the IP address of the device that has the MAC address of the entry.

The *mac-addr* parameter specifies the MAC address of the entry.

The e**thernet** *slot/port* variable specifies the port number attached to the device that has the MAC address of the entry.

Displaying ARP inspection information

You can display ARP inspection information using the **show ip arp-inspection** and the **show ip static-arp** commands as shown in the following.

Displaying ARP inspection status and ports

To display the ARP inspection status for a VLAN and the trusted/untrusted ports in the VLAN, enter the following command.

```
device# show ip arp-inspection
ARP inspected VLANs:
1000
ARP inspection trusted ports:
ethe 2/1
```

Syntax: show ip arp-inspection [**vlan vlan_id**]

The **vlan** *vlan_id* parameter specifies the ID of a configured VLAN.

Displaying ARP inspection statistics

You can use the **show ip arp-statistics** command to display ARP inspection counters for all ports on the device, as shown in the following.

```
device# show ip arp-inspection-statistics
Module 1:
Port     Arp Packets Captured      Arp Packets Failed Inspection
1/1         0                        0
1/2         0                        0
1/3         0                        0
1/4         0                        0
1/5         0                        0
1/6         0                        0
1/7         0                        0
1/8         0                        0
1/9         0                        0
1/10        0                        0
1/11        0                        0
1/12        0                        0
1/13        0                        0
1/14        0                        0
1/15        0                        0
1/16        0                        0
1/17        0                        0
1/18        0                        0
1/19        0                        0
1/20        0                        0
Module 3:
Port     Arp Packets Captured      Arp Packets Failed Inspection
3/1         0                        0
3/2         0                        0
3/3         0                        0
3/4         690                      153
```

Specifying a port number with the **show ip arp-statistics** command displays the statistics for that port only, along with details of the last five ARP packets that failed inspection, as shown in the following.

```
device# show ip arp-inspection-statistics ethernet 3/4
Arp packets captured: 695
Arp packets failed inspection: 158
Last 5 packets failed inspection:
Time                 Op  Target IP Target Mac     Source IP Source Mac     Vlan
2007-10-24  18:53:28 2   10.1.1.1 000c.dbe2.9353 10.1.1.2 0000.0900.0005 1
2007-10-24  18:53:29 2   10.1.1.1 000c.dbe2.9353 10.1.1.2 0000.0900.0005 1
2007-10-24  18:53:30 2   10.1.1.1 000c.dbe2.9353 10.1.1.2 0000.0900.0005 1
2007-10-24  18:53:32 2   10.1.1.1 000c.dbe2.9353 10.1.1.2 0000.0900.0005 1
2007-10-24  18:53:33 2   10.1.1.1 000c.dbe2.9353 10.1.1.2 0000.0900.0005 1
```

Syntax: show ip arp-inspection-statistics [**slot slot-num** | **ethernet slot/port**]

The **slot** option allows you to limit the display of ARP inspection statistics to the Ethernet interface module in the slot specified by the *slot-num* variable.

The **ethernet** option allows you to limit the display of ARP inspection statistics to the port specified by the *slot/port* variable. It also provides details of the last five ARP packets received by the specified port that failed inspection.

This display shows the following information.

TABLE 83 Show ip arp-inspection-statistics

This field...	Displays...
Port	The slot/port number.
Arp packets captured	The number of ARP packets captured for the specified port.
Arp packets failed inspection	The number of captured ARP packets that failed inspection for the specified port.
The following fields apply to the first five packets that failed inspection on the specified port .	
Time	The date and time that the packet was received on the port.
Op	The ARP operation mode.
Target IP	The destination IP address of the ARP rejected packet.
Target MAC	The destination MAC address of the ARP rejected packet.
Source IP	The source IP address of the ARP rejected packet.
Source MAC	The source MAC address of the ARP rejected packet.
VLAN	The VLAN number of the ARP rejected packet.

Clearing ARP inspection counters

You can use the clear arp-inspection-statistics command to clear the ARP inspection statistics counters for all ports on the device or for a specified module or port as shown in the following.

```
clear arp-inspection-statistics ethernet 3/1
```

Syntax: clear ip arp-inspection-statistics [**slot slot-num** | **ethernet slot/port**]

The **slot** option allows you to clear ARP inspection statistics for a single Ethernet interface module in a slot specified by the *slot-num* variable.

The **ethernet** option allows you to clear ARP inspection statistics for a single port specified by the *slot/port* variable.

Displaying the ARP table

To display the ARP Inspection table, enter the following command.

```
device# show ip static-arp
Total no. of entries: 4
  Index   IP Address          MAC Address         Port      VLAN  ESI
  1       10.1.1.1             0001.0001.0001      1/1
  2       10.6.6.2             0002.0002.0002      1/2
  3       10.6.6.7             1111.1111.1111      2/1...
          Ports  : ethe 2/1 to 2/7 ethe 3/1 to 3/2
  4       10.7.7.7             0100.5e42.7f40      3/3
```

The command displays all ARP entries in the system. The illustration above shows the output from a Brocade NetIron XMR Series and Brocade NetIron MLX Series device and includes a multi-port static ARP entry.

Syntax: show ip static-arp

DHCP snooping

NOTE
DHCP snooping supports only IPv4 traffic.

Dynamic Host Configuration Protocol (DHCP) snooping enables the device to filter untrusted DHCP packets in a subnet. DHCP snooping prevents MiM attacks, such as a malicious user posing as a DHCP server sending false DHCP server reply packets with the intention of misdirecting other users. DHCP snooping can also stop unauthorized DHCP servers and prevent errors due to user mis-configuration of DHCP servers.

NOTE
DHCP snooping does not dynamically build the ARP Inspection table.

How DHCP snooping works

When enabled on a VLAN, DHCP snooping stands between untrusted ports (those connected to host ports) and trusted ports (those connected to DHCP servers). A VLAN with DHCP snooping enabled forwards DHCP request packets from clients and discards DHCP server reply packets on untrusted ports, and it forwards DHCP server reply packets on trusted ports to DHCP clients, as shown in the following figures.

FIGURE 187 DHCP snooping at work - on untrusted port

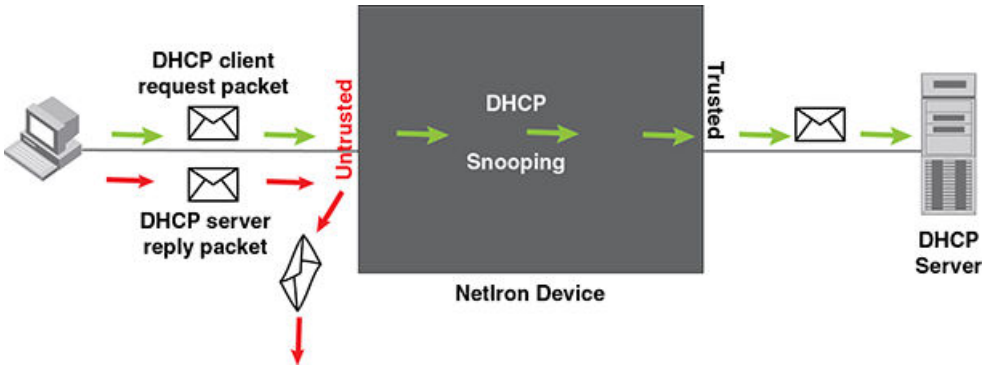
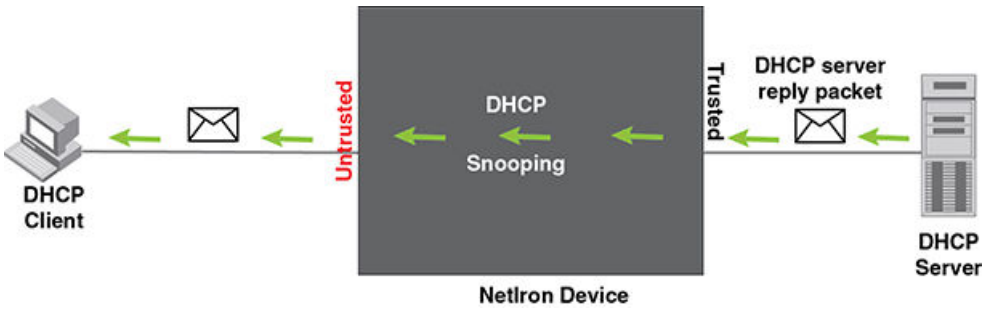


FIGURE 188 DHCP snooping at work - on trusted port



DHCP binding database

On trusted ports, DHCP server reply packets are forwarded to DHCP clients. The DHCP server reply packets collect client IP to MAC address binding information, which is saved in the DHCP binding database. This information includes MAC address, IP address, lease time, VLAN number, and port number.

In the Brocade device configuration, the DHCP binding database is integrated with the enhanced ARP table, which is used by Dynamic ARP Inspection. For more information, refer to ARP entries on page 719.

The lease time will be refreshed when the client renews its IP address with the DHCP server; otherwise the device removes the entry when the lease time expires.

System reboot and the binding database

To allow DAI and DHCP snooping to work smoothly across a system reboot, the binding database is saved to a file in the system flash memory after the user issues the "reload" command. DHCP learned entries are written to the system flash memory before the device reboots. The flash file is written and read only if DHCP snooping is enabled.

Configuring DHCP snooping

Follow the steps listed below to configuring DHCP snooping.

1. Enable DHCP snooping on a VLAN.
2. For ports that are connected to a DHCP server, change their trust setting to trusted.

   The following table shows the default settings of DHCP snooping:

Feature	Default
DHCP snooping	Disabled
Trust setting for ports	Untrusted

Enabling DHCP snooping on a VLAN

DHCP packets for a VLAN with DHCP snooping enabled are inspected.

DHCP snooping is disabled by default. This feature must be enabled on the client and the DHCP server VLANs. To enable DHCP snooping, enter the following global command for these VLANs.

```
device(config)#ip dhcp-snooping vlan 2
```

The command enables DHCP snooping for a VLAN or a range of VLANS.

Syntax: [no] ip dhcp-snooping vlan vlan-number [**to vlan_number**] [**insert-relay-information**]

The *vlan-number* variable specifies the ID of a configured client or DHCP server VLAN.

If the [insert-relay-information] option is enabled, then DHCP option 82 is inserted in all the DHCP request packets. Refer to DHCP binding database on page 727 for more information.

DHCP snooping suboptions

When the DHCP relay agent information option is enabled, the DHCP relay adds the option 82 information to packets it receives from clients, then forwards the packets to the DHCP server. The DHCP server uses the option 82 information to decide which IP address to assign to the client or the DHCP server may use the information in the option 82 field for determining which services to grant to the

client. The DHCP server sends its reply back to the DHCP relay, which removes the option 82 information field from the message, and then forwards the packet to the client.

Option 82 information is made up of a series of suboptions. Brocade supports suboption 1, suboption 2, and suboption 9.

- Agent Circuit ID (suboption 1) --An ASCII string identifying the interface on which a client DHCP packet is received.
- Agent Remote ID (suboption 2) --An ASCII string assigned by the relay agent that securely identifies the client.
- Vendor-Specific (suboption 9) --Contains the Internet Assigned Numbers Authority (IANA) enterprise number (4874) and the layer 2 circuit ID and the user packet class.

Suboption 1 and suboption 2 are usually determined by the client network access device and depend on the network configuration.

Suboption 9 can be used to associate specific data with the DHCP messages relayed between the DHCP relay and the DHCP server. The suboption 9 can include the client's IEEE 802.1p value, which identifies the client's user priority.

```
device(config)# ip dhcp-snooping vlan 10 insert-relay-information apend
```

Syntax: [no] ip dhcp-snooping vlan vlan-number insert-relay-information [append *type*]

The *vlan-number* variable specifies the ID of a configured client or DHCP server VLAN.

Use the **append** option to add suboption 9 in the option82 field of the DHCP request as per RFC4342 before forwarding to server.

The *type* options to specify the sub option's type (suboption 1 and suboption 2). You can add a suboption with a specific number. The range is between 3 and 254.

Enabling trust on a port

The default trust setting for a port is untrusted. To enable trust on a port connected to a DHCP server, enter commands such as the following.

```
device(config)#interface ethernet 1/1
device(config-if-e10000-1/1)#dhcp-snooping-trust
```

Port 1/1 is connected to a DHCP server. The commands change the CLI to the interface configuration level of port 1/1 and set the trust setting of port 1/1 to trusted.

Syntax: [no] dhcp-snooping-trust

Clearing the DHCP binding database

You can clear the DHCP binding database using the **clear dhcp-binding** command.

You can remove all entries in the database, or remove entries for a specific IP subnet, a VRF instance, or a VLAN id.

To remove all entries from the DHCP binding database, enter the following command.

```
device# clear dhcp-binding
```

For example, to clear entries for a specific IP subnet, enter a command such as the following.

```
device# clear dhcp 10.10.102.4
```

Syntax: clear dhcp [**ip subnet**] [**vlan vlan_id**] [**vrf vrf_name**]

The *vlan_id* variable specifies the ID of a configured VLAN.

The *vrf_name* variable specifies the VRF instance.

DHCP option 82 insertion

DHCP option 82 insertion can be used to assist DHCP servers to implement dynamic address policy. When DHCP option 82 is present in DHCP packets, DHCP servers gets additional information about the clients' identity.

The Brocade device inserts DHCP option 82 when relaying DHCP request packets to DHCP servers. When DHCP server reply packets are forwarded back to DHCP clients, and sub-option 2 matches the local port MAC address, then DHCP option 82 is deleted. The vlan/port information is used to forward the DHCP reply. Refer to the following figures:

FIGURE 189 DHCP option 82 is added to the packet

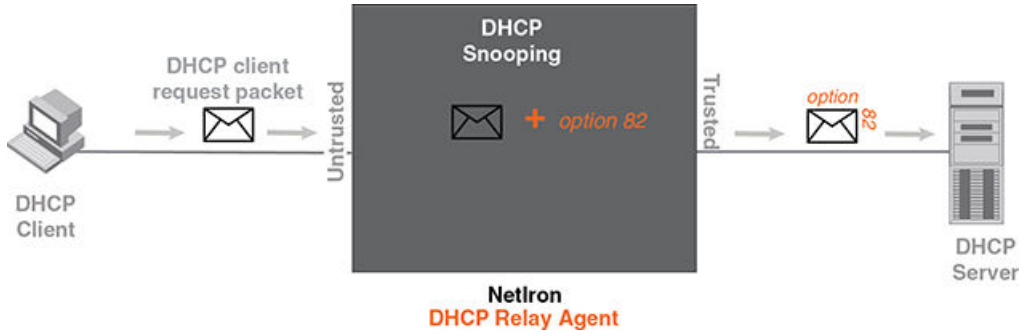
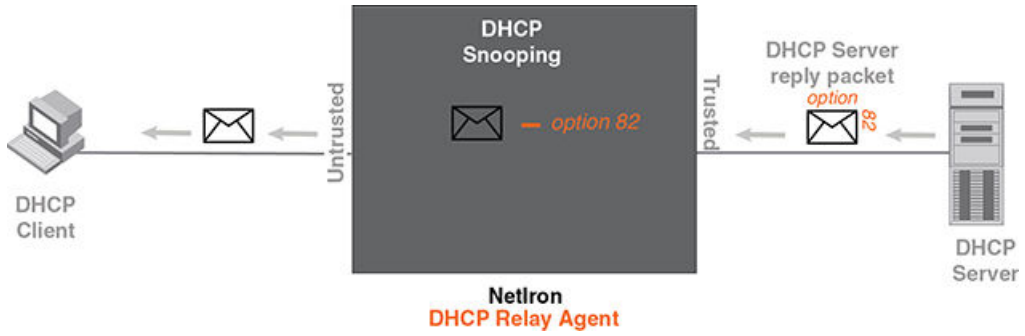


FIGURE 190 DHCP option 82 is removed from the packet



The option 82 insertion/deletion feature is available only when DHCP snooping is enabled for the client/server ports, and when the device is configured as a DHCP relay agent. By default, DHCP option 82 is off.

DHCP option 82 contains two sub-options; sub-option 1 (circuit ID) and sub-option 2 (remote ID).

Sub-option 1, relay agent circuit ID is in the following format.

VLAN id (2 bytes) / module id (1 byte) / port id (1 byte) (The module and port id will be 1 based).

The circuit ID identifies the location of the port, showing where the DHCP request comes from.

Typical address allocation is based on the gateway address of the relay agent.

Sub-option 2, Remote ID is in the following format.

Brocade NetIron XMR Series base MAC address (6 bytes)

Displaying DHCP snooping status and ports

To display the DHCP snooping status for a VLAN and the trusted and untrusted ports in the VLAN, enter the following command.

```
device#show ip dhcp-snooping vlan 172
IP DHCP snooping VLAN 172: Enabled
  Trusted Ports : ethe 5/2 ethe 5/4
  Untrusted Ports : ethe 4/24 ethe 9/4 to 9/5 ethe 9/12 ethe 9/14
```

Syntax: show ip dhcp-snooping [**vlan vlan-id**]

Displaying DAI binding entries

To display all ARP inspection binding entries, including dhcp bindings specific to a VRF instance, enter the following command.

```
device(config)#show dai 10.1.1.0/24
Total no. of entries: 51
Idx Type IP Address      MAC Address       Port    Vlan Server IP    LTime
1   D    10.1.1.19       aabb.cc00.0012     10      10.1.1.2       3360
2   D    10.1.1.22       aabb.cc00.0007     10      10.1.1.2       3360
3   D    10.1.1.25       aabb.cc00.0030     10      10.1.1.2       3360
4   D    10.1.1.26       aabb.cc00.0004     10      10.1.1.2       3360
5   D    10.1.1.30       0030.488a.1c25     10      10.1.1.2       40200
6   D    10.1.1.32       aabb.cc00.0001     10      10.1.1.2       3360
7   D    10.1.1.34       aabb.cc00.0019     10      10.1.1.2       1560
8   D    10.1.1.39       aabb.cc00.000d     10      10.1.1.2       3360
9   D    10.1.1.44       aabb.cc00.0020     10      10.1.1.2       3360
10  D    10.1.1.46       aabb.cc00.0022     10      10.1.1.2       3360
```

Syntax: show dai [**vrf vrf_name**] [**vlan vlan_id**] [**ip-subnet**]

The *vrf_name* variable specifies the ARP entries that belong to a given VRF instance.

The *vlan_id* variable specifies the ID of a configured VLAN.

The *ip subnet* variable specifies the ARP entries that belong to specific IP-subnet address.

The following table describes the parameters of the **show dai** command:

TABLE 84 Display of show dai

This field...	Displays...
Index (Idx)	The row number of this entry in the IP route table.
Type	The ARP entry type, which can be any one of the following: Dynamic - The Layer 3 Switch learned the entry from an incoming packet. Static - The Layer 3 Switch loaded the entry from the static ARP table when the device for the entry was connected to the Layer 3 Switch. DHCP - The Layer 3 Switch learned the entry from the DHCP binding address table.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The source port for the host vlan.
Vlan Server IP	The VLAN Server IP address of the server which assigns the IP/MAC mapping.
LTime	The lease time (aging timer for a DHCP entry).

Displaying DHCP snooping statistics counters

NOTE
The last five dropped packets are displayed through the CLI. Notifications and traps are not sent.

To display the DHCP snooping statistics counters, enter the following command.

```
device#show ip dhcp-snooping-statistic slot 1
Module 1:
Port     DHCP Packets Captured      DHCP Packets dropped
1/1         0                        0
1/2         0                        0
1/3         0                        0
1/4         0                        0
1/5         0                        0
1/6         0                        0
1/7         0                        0
1/8         0                        0
1/9         0                        0
1/10        0                        0
1/11        0                        0
1/12        0                        0
1/13        0                        0
1/14        0                        0
1/15        0                        0
1/16        0                        0
1/17        0                        0
1/18        0                        0
1/19        9                        0
1/20        8                        6
```

The following table describes the output of the show ip dhcp snooping statistic.

TABLE 85 Output from the show ip dhcp snooping statistic slot

This field...	Displays...
Module	Module number as positioned in the chassis.
Port	Port number in specified module.
DHCP Packets Captured	The number of DHCP packets captured on the port.
DHCP Packets Dropped	The number of DHCP packets dropped by DHCP snooping.

To display the DHCP snooping statistics counters for Ethernet ports, enter the following command.

```
device#show ip dhcp-snooping-statistic eth 1/20
DHCP packets captured: 9
DHCP packets dropped by snooping: 7
Last 5 packets dropped by snooping:
Time                 DHCP type Source Mac/     Server IP/      Vlan
                               Source IP       Gateway IP
2008-05-03  00:29:43 OFFER     0030.4843.37ad  10.1.1.125     11
                               10.1.1.125     10.1.1.10
2008-05-03  00:29:59 OFFER     0030.4843.37ad  10.1.1.125     11
                               10.1.1.125     10.1.1.10
2008-05-03  00:31:18 OFFER     0030.4843.37ad  10.1.1.125     11
                               10.1.1.125     10.1.1.10
2008-05-03  00:31:22 OFFER     0030.4843.37ad  10.1.1.125     11
                               10.1.1.125     10.1.1.10
2008-05-03  00:31:30 OFFER     0030.4843.37ad  10.1.1.125     11
                               10.1.1.125     10.1.1.10
```

Syntax: show ip dhcp-snooping-statistic [**slot slot**] | [**ethernet slot/port**]

NOTE
If an Ethernet port is provided, the last five dropped packets are displayed

1. DHCP packets captured
2. The number of DHCP packets captured on port 20.
3. DHCP packets dropped by snooping
4. The number of DHCP packets dropped by DHCP snooping.
5. Last 5 packets dropped by snooping
6. The last 5 DHCP packets dropped per port.
7. Time
8. The time tracking system for collecting statistically information. Date and time are displayed.
9. DHCP Type
10. The DHCP Type displays the following: OFFER - When the server responds with a proposal of parameters. ACK- When the server assign an IP address. NAK- When the server rejects the request from the client.
11. Source MAC or Source IP
12. The Source MAC or Source IP address
13. Server IP or Gateway IP
14. The Serve IP or Gateway IP
15. Vlan
16. The VLAN number that DHCP Snooping was rejected on.

Clearing DHCP snooping counters

To clear the DHCP snooping statistic counters for a specific slot, enter the following command.

```
device#clear dhcp-snooping-statistics slot 1
```

To clear the DHCP snooping statistic counters for a specific ethernet port, enter the following command.

```
device#clear dhcp-snooping-statistics ethernet 1/20
```

Syntax: clear dhcp-snooping-statistics [**slot slot**] | [**ethernet slot/port**]

DHCP snooping configuration example

The following example configures VLAN 2 and VLAN 20, and changes the CLI to the global configuration level to enable DHCP snooping on the two VLANs. The commands are as follows.

```
device(config)#vlan 2
device(config-vlan-2)#untagged ethe 1/3 to 1/4
device(config-vlan-2)#router-interface ve 2
device(config-vlan-2)#exit
device(config)# ip dhcp-snooping vlan 2
device(config)#vlan 20
device(config-vlan-20)#untagged ethe 1/1 to 1/2
device(config-vlan-20)#router-interface ve 20
device(config-vlan-20)#exit
device(config)#ip dhcp-snooping vlan 20
```

On VLAN 2, client ports 1/3 and 1/4 are untrusted by default: all client ports are untrusted. Hence, only DHCP client request packets received on ports 1/3 and 1/4 are forwarded.

On VLAN 20, ports 1/1 and 1/2 are connected to a DHCP server. DHCP server ports are set to trusted.

```
device(config)#interface ethernet 1/1
device(config-if-e1000-1/1)#dhcp-snooping-trust
device(config-if-e1000-1/1)#exit
device(config)#interface ethernet 1/2
device(config-if-e1000-1/2)#dhcp-snooping-trust
device(config-if-e1000-1/2)#exit
```

Hence, DHCP sever reply packets received on ports 1/1 and 1/2 are forwarded, and client IP or MAC binding information is collected.

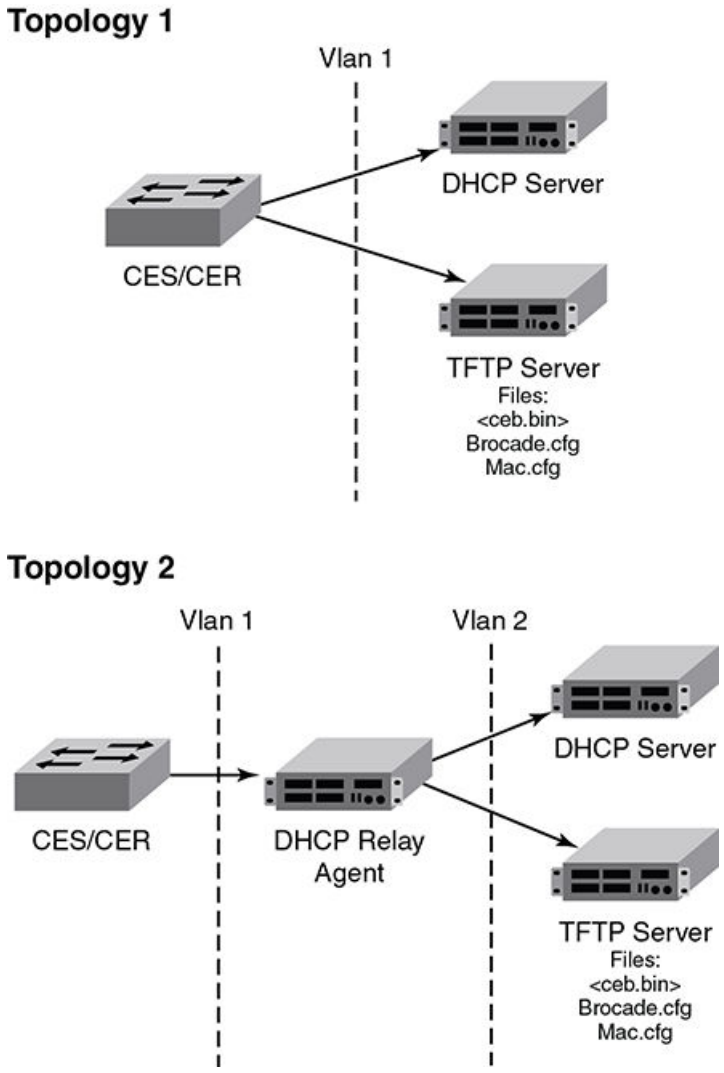
The example also sets the DHCP server address for the local relay agent.

```
device(config)# interface ve 2
device(config-vif-2)#ip address 10.20.20.1/24
device(config-vif-2)#ip helper-address 10.30.30.4
device(config-vif-2)#interface ve 20
device(config-vif-20)#ip address 10.30.30.1/24
```

Zero Touch Provisioning

Zero Touch Provisioning allows Brocade devices to automatically obtain a dynamically assigned DHCP address, negotiate address lease renewal, and download flash image and configuration files. Figure 194 provides information about this feature.

FIGURE 191 Zero Touch Provisioning



Zero Touch Provisioning consists of the following steps.

1. The IP address validation and lease negotiation enables the DHCP client to automatically obtain and configure an IP address.
   a) As the Brocade device comes online, it checks if the DHCP client is enabled on any of the data ports.
   b) If no data port is enabled, the device tries to obtain an address on the management port.

      NOTE
      The management port is not enabled by default and needs to be enabled manually for the feature to operate.If the management port is configured with a static IP address, the Zero Touch Provisioning feature is automatically disabled.
2. The TFTP flash image is downloaded and updated. The device compares the file names of the requested flash image and the image stored in flash memory. If the names are different, the device downloads the new image from a TFTP server and writes the downloaded image to flash memory.

3. The Zero Touch Provisioning feature supports update of the existing monitor image and reload of the device.
4. The device downloads configuration files from a TFTP server and saves them as the running configuration.

Zero Touch Provisioning limitations

The following limitations apply to the Zero Touch Provisioning feature.

- By default, Zero Touch Provisioning is always enabled on a management port.
- Zero Touch Provisioning fails on a management port which has a static address configured on it.
- Zero Touch Provisioning does not support trunked ports or Link Aggregation Control Protocol (LACP) ports.
- During the Zero Touch Provisioning update, the existing configuration takes precedence over any configuration downloaded from the TFTP server.
- VE and VLAN numbers that are chosen for Zero Touch Provisioning cannot be used for other configurations.

Upgrade and downgrade considerations

- During a network upgrade procedure, the downloaded configuration files (as a part of the Zero Touch Provisioning process) may contain commands that cannot be executed using the current software version. In such a scenario, download the configuration files after a system reboot following the image download.
- During a network downgrade procedure, inspect the running configuration, because the system ignores errors due to incompatible commands from the previous configuration.

Supported options for DHCP

Zero Touch Provisioning supports the following DHCP options:

- DHCP Parameter Request List
  - Subnet Mask
  - Domain Name
  - Router
  - Host Name (optional)
  - TFTP Server Name
- DHCP Client
  - Server ID
  - IP Address Lease
  - Renewal Time Value
  - Rebind Time value
  - Subnet Net mask
  - Domain Name
  - Router
  - Domain Server
  - Host Name
  - TFTP Server Name

Supported messages for DHCP servers

Zero Touch Provisioning supports the following DHCP messages:

- DHCPACK
- DHCPDECLINE
- DHCPDISCOVER
- DHCPNAK
- DHCPOFFER
- DHCPRELEASE
- DHCPREQUEST

NOTE
Zero Touch Provisioning does not support the DHCPINFORM message.

Configuring Zero Touch Provisioning

Zero Touch Provisioning allows a Brocade device to dynamically update its running configuration. This feature uses the DHCP client for address allocation and the TFTP server to download a specific configuration file.

NOTE
Virtual Ethernet (VE) and virtual LAN (VLAN) have to exist for the DHCP configuration to work.

To enable Zero Touch Provisioning on a port, use the **ip dhcp-client vlan** command. This command enables the autoconfiguration on any in-band port with the DHCP client configured on it.

```
device(config)# ip dhcp-client vlan 227 ve 227 tagged 1/1 auto-update enabled
```

Syntax: [no] ip dhcp-client vlan *vlannumber* **ve** [*venumber* | **tagged** | **untagged** | *slot/port* | **auto-update enabled** | **auto-update disabled**]

The *vlan number* variable is the desired VLAN number for sending out tagged or untagged DHCP requests.

The *ve number* variable is the router interface number for sending out tagged or untagged DHCP requests.

The **tagged** or **untagged** options add the port as a tagged or untagged member of the VLAN.

The *slot/port* variable is the desired slot and port for the DHCP client.

The **auto-update enabled** option enables autoconfiguration on the port.

The **auto-update disabled** option disables autoconfiguration on the port.

NOTE
VLAN and VE are created when you run the **ip dhcp-client vlan** command.

Disabling auto-update on a port

To disable only the auto-update feature on a port, enter the following command.

```
device(config)# ip dhcp-client default-vlan untagged 1/1 auto-update disabled
```

Syntax: ip dhcp-client vlan *vlannumber* **ve** [*venumber* | **tagged** | **untagged** | *slot/port* | **auto-update disabled**]

Enabling autoconfiguration on a default VLAN

To enable autoconfiguration on a default VLAN, use the following command when auto-update is enabled on port 1/1.

```
device(config)# ip dhcp-client default-vlan untagged 1/1 auto-update enabled
```

Enabling autoconfiguration on a tagged VLAN

To enable autoconfiguration on a tagged VLAN, use the following command when the VLAN number is 227, the VE number is 227, and auto-update is enabled on port 1/1.

```
device(config)# ip dhcp-client vlan 227 ve 227 tagged 1/1 auto-update enabled
```

Enabling autoconfiguration on an untagged VLAN

To enable autoconfiguration on an untagged VLAN, use the following command when the VLAN number is 227, the VE number is 227, and auto-update is enabled on port 1/1.

```
device(config)# ip dhcp-client vlan 227 ve 227 untagged 1/1 auto-update enabled
```

Enabling autoconfiguration on a management port

To enable autoconfiguration on a management port, use the following command.

```
device(config)# ip dhcp-client vlan
```

Use the **no ip dhcp-client vlan** command to disable the DHCP client and autoconfiguration for the designated port.

Displaying Zero Touch Provisioning information

Run the **show ip** and the **show ip interface** commands to display information about the successful implementation of Zero Touch Provisioning.

```
device#show ip
Global Settings
IP CAM Mode: static IPVPN CAM Mode: static
  ttl: 64, arp-age: 10, bootp-relay-max-hops: 4, icmp-error-rate: 400
  IP Router-Id: 10.1.1.1
DHCP server address: 10.21.96.1
TFTP server address: 10.21.96.1
Configuration filename: brocade.cfg
enabled: UDP-Broadcast-Forwarding  ICMP-Redirect  Source-Route  Load-Sharing  RARP
disabled: Directed-Broadcast-Forwarding  drop-arp-pending-packets  IRDP  Proxy-ARP  RPF-Check  RPF-Exclude-
Default  RIP  BGP4 IS-IS OSPF  VRRP  VRRP-Extended  VSRP
Configured Static Routes: 2

device#show ip interface
Interface  IP-Address     OK?  Method   Status       Protocol   VRF          Type     Lease Time
eth 1/1    10.1.1.1        YES  NVRAM    admin/down   down       default-vrf  Static   N/A
mgmt 1     10.21.96.160   YES  NVRAM    up           up         default-vrf  Dynamic  672651
```

Table 86 describes the fields from the output of **show ip interface** command.

TABLE 86 Output display of show ip interface command

Field	Description
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.

TABLE 86 Output display of show ip interface command (continued)

Field	Description
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI or Web Management Interface, but have not saved the configuration, the entry for the interface in the Method field is "manual".
Status	The link status of the interface. If you have disabled the interface with the **disable** command, the entry in the Status field will be "administratively down". Otherwise, the entry in the Status field will be either "up" or "down".
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the Protocol field will be "up". Otherwise, the entry in the Protocol field will be "down".
VRF	The VRF type.
Type	The type of lease.
Lease Time	The time when this lease will expire.

IP source guard

IP Source Guard permits traffic from only valid source IP addresses. IP source guard is used on client ports to prevent IP source address spoofing. Generally, IP source guard is used together with DHCP snooping and Dynamic ARP Inspection on untrusted ports. Refer to DHCP snooping on page 726 and Dynamic ARP inspection on page 718.

When IP Source Guard is first enabled, only DHCP packets are allowed and all other IP traffic is blocked. IP Source Guard uses IP or MAC bindings inside the ARP Inspection table to detect a valid IP address. When the system learns a valid IP address on the port, the client port then allows IP traffic to enter. If the source IP address of a packet does not match any of the IP addresses inside the ARP Inspection table, the packet is dropped. Only traffic with valid source IP addresses are permitted.

The system learns of a valid IP address from ARP. For information on how the ARP table is populated, Refer toARP entries on page 719.

Enabling IP source guard

The **source-guard** command sets a port as an IP Source Guarded port. DHCP Snooping should be configured before you enable the IP Source Guard feature.

The default setting is disabled. To enable a port as an IP Source Guarded port, enter the following commands.

```
device(config)# interface ethernet 2/2
device(config-if-e10000-2/2)# source-guard
```

The commands change the CLI to the interface configuration level for port 2/2 and enable IP source guard on the port.

Syntax: [no] source-guard

> NOTE
> When IP Source Guard is enabled on a port it must have the same configuration as the primary port, otherwise it will not implemented as IP Source Guarded.

Enabling IP source inspection on a VLAN

IP Source Guard configuration is enabled on ports per vlan. When IP Source Guard is enabled on a vlan, by default all ports inside the vlan are set as "unguarded". You can selectively turn on which ports inside the vlan to be set as "guarded". Initially, when the vlan port is IP

Source Guarded, only DHCP packets are allowed to get through. However, as IP or MAC binding is learned from DHCP snooping, or if it is manually configured, only packets with valid source IP address are allowed through.

There are two modes for IP Source Guard; strict mode and loose mode. You can configure either strict or loose mode during IP Source Guard vlan configuration. In a strict mode, the IP source address is bound to a particular port and vlan. Only packets with an IP address coming from a particular vlan port is considered valid. If the same source IP address is coming from a different port, then it is considered an attack and is dropped. The strict mode provides more security, but it does not allow for a layer 2 occurrence in a vlan. In a loose mode, the IP source address is bound to a vlan. Only packets with IP source addresses that come from ports within the vlan are considered valid.

To enable IP Source Inspection for a VLAN or a range of VLANs, enter the following command.

```
device(config)# ip source-inspection vlan 2
```

Syntax: [no] ip source-inspection vlan vlan_number [**to vlan_number**] [**strict**]

The source IP addresses for VLAN IP packets are inspected for any port when IP Source Guard is enabled.

The *vlan_number* variable specifies the ID of a configured VLAN.

If the strict option is enabled, then valid IP source address is bound to a particular source port. This configuration can be learned from a DHCP reply, or manually configured.

> NOTE
> The strict mode requires DHCP relay-information insertion to be turned on.

Displaying IP source inspection status and ports

To display the IP Source Guard status for a VLAN, and the guarded or unguarded ports in the VLAN, enter the following command.

```
device(config)#sh ip source-inspection vlan 10
IP Source Inspection configuration for VLAN 10:
Inspection mode: loose
un-guarded ports:
 ethe 1/4 ethe 1/18
guarded ports:
 ethe 1/20
```

The **show ip source-inspection vlan** command displays IP Source inspection configuration for VLAN 10 in loose mode.

Syntax: show ip source-inspection [**vlan vlan_id**]

The *vlan_id* variable specifies the ID of a configured vlan.

> NOTE
> This command is also available for debugging purposes on the Interface Module.

IP source guard CAM

The Brocade device configuration uses a layer 4 ACL CAM to implement IP Source guard. When IP or MAC binding is learned or configured on an IP Source Guarded vlan-port, a layer 4 ACL CAM is programmed to allow valid source IP addresses.

When ACL is manually configured, a configuration conflict occurs with IP Source Guard, because it uses a layer 4 ACL CAM. The Brocade device gives user ACL configuration a higher priority. When both IP Source Guard and user ACL is configured, the user ACL configuration takes precedence over IP Source Guard.

IP Source Guard uses layer 4 ACL CAM to check layer 2 switched traffic. When IP Source Guard is configured, the layer 3 port check flag is turned on. When IP Source Guard is configured, all traffic from the same physical port is subject to a layer 4 ACL check.

Configuring IP source guard CAM partition

IP Source Guard creates two CAM sub-partitions. The CAM sub-partitions include IP_SOURCE_GUARD_PERMIT and IP_SOURCE_GUARD_DENY. All CAM entries that are permitted, go to the IP_SOURCE_GUARD_PERMIT sub-partition. All CAM entries that are denied, go to the IP_SOURCE_GUARD_DENY sub-partition. The **system-max ip-source-guard-cam** command allows you to control the size of both IP_SOURCE_GUARD_PERMIT and IP_SOURCE_GUARD_DENY sub-partitions.

To specify a partition size of the IP Source Guard CAM, enter the following command.

```
device(config)#system-max ip-source-guard-cam 1008
```

Syntax: [no] system-max [**ip-source-guard-cam decimal**]

By default, **no** system-max is configured.

The *decimal* variable specifies the range that is supported for configuring IP Source Guard CAM sub-partitions. The decimal range is from 0 to 131072. The default is 0.

Configuring forwarding parameters

The following configurable parameters control the forwarding behavior of the Brocade device:

- Time-To-Live (TTL) threshold
- Forwarding of directed broadcasts
- Forwarding of source-routed packets
- Ones-based and zero-based broadcasts

All these parameters are global and thus affect all IP interfaces configured on the Brocade device.

To configure these parameters, use the procedures in the following sections.

Changing the TTL threshold

The TTL threshold prevents routing loops by specifying the maximum number of router hops an IP packet originated by the Brocade device can travel through. Each device capable of forwarding IP that receives the packet decreases the packet's TTL by one. If a device receives a packet with a TTL of 1 and reduces the TTL to zero, the device drops the packet.

The default TTL is 64. You can change the TTL to a value from 1- 255.

To modify the TTL threshold to 25, enter the following commands.

```
device(config)# ip ttl 25
```

Syntax: [no] ip ttl 1-255

Enabling forwarding of directed broadcasts

A directed broadcast is an IP broadcast to all devices within a single directly-attached network or subnet. A net-directed broadcast goes to all devices on a given network. A subnet-directed broadcast goes to all devices within a given subnet.

> NOTE
> A less common type, the all-subnets broadcast, goes to all directly-attached subnets. Forwarding for this broadcast type also is supported, but most networks use IP multicasting instead of all-subnet broadcasting.

Forwarding for all types of IP directed broadcasts is disabled by default. You can enable forwarding for all types if needed. You cannot enable forwarding for specific broadcast types.

To enable forwarding of IP directed broadcasts, enter the following command.

```
device(config)# ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

The software makes the forwarding decision based on the device'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.

```
device(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.

```
device(config)# interface ethernet 1/1
device(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 Brocade device supports both types of IP source routing:

- **Strict source routing** - requires the packet to pass through only the listed routers. If the Brocade device receives a strict source-routed packet but cannot reach the next hop interface specified by the packet, the Brocade device discards the packet and sends an ICMP Source-Route-Failure message to the sender.

  NOTE
  The Brocade device allows you to disable sending of the Source-Route-Failure messages. Refer to Disabling ICMP messages on page 743.

- **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 Brocade device forwards both types of source-routed packets by default. You cannot enable or disable strict or loose source routing separately.

To disable forwarding of IP source-routed packets, enter the following command.

```
device(config)# no ip source-route
```

Syntax: [no] ip source-route

To re-enable forwarding of source-routed packets, enter the following command.

```
device(config)# ip source-route
```

Enabling support for zero-based IP subnet broadcasts

By default, the Brocade device treats IP packets with all ones in the host portion of the address as IP broadcast packets. For example, the Brocade device treats IP packets with 10.157.22.255/24 as the destination IP address as IP broadcast packets and forwards the packets to all IP hosts within the 10.157.22.x subnet (except the host that sent the broadcast packet to the Brocade device).

Most IP hosts are configured to receive IP subnet broadcast packets with all ones in the host portion of the address. However, some older IP hosts instead expect IP subnet broadcast packets that have all zeros instead of all ones in the host portion of the address. To

accommodate this type of host, you can enable the Brocade device to treat IP packets with all zeros in the host portion of the destination IP address as broadcast packets.

NOTE
When you enable the Brocade device for zero-based subnet broadcasts, the Brocade device still treats IP packets with all ones the host portion as IP subnet broadcasts too. Thus, the Brocade device can be configured to support all ones only (the default) or all ones and all zeroes.

NOTE
This feature applies only to IP subnet broadcasts, not to local network broadcasts. The local network broadcast address is still expected to be all ones.

To enable the Brocade device for zero-based IP subnet broadcasts in addition to ones-based IP subnet broadcasts, enter the following command.

```
device(config)# ip broadcast-zero
```

Syntax: [**no**] **ip broadcast-zero**

Allowing multicast addresses as source IP addresses

By default packets with multicast addresses as source IP address are dropped at the packet processor in the line card. You can now disable the dropping of packets with multicast addresses as source IP address.

Unicast or multicast destination IP address forwarding works as usual, regardless of whether you enable or disable this feature. You can allow multicast addresses as source IP address for all packets or switched traffic packets only. Packets with class D and E addresses as source IPv4 address and packets with prefixes beginning with 0xFF as source IPv6 addresses (for example FF01::11), are also allowed once you enable this feature.

NOTE
Unicast Reverse Path Forwarding is disabled once you allow multicast addresses as source IP addresses.

Perform the following steps to allow multicast addresses as source IP addresses.

1. Enter global configuration mode.
2. To allow multicast addresses as source IP addresses enter the **ip allow-src-multicast** command followed by the options *decimal* or **all**.
   The following example allows multicast addresses as source IP address for all traffic.

   ```
   device(config)# ip allow-src-multicast all
   ```

3. To allow multicast addresses as source IP address for only switched traffic, enter the **ip allow-src-multicast switched-traffic** command followed by the options *decimal* or **all**.
   The following example allows multicast addresses as source IP address for switched traffic on a specific slot.

   ```
   device(config)# ip allow-src-multicast switched-traffic 3
   ```

4. To view if the disable packet drop for multicast IPv4 or IPv6 as source IP is enabled or disabled for switched-traffic only, use the **show ip allow-src-multicast switched-only** command.
5. To view if the disable packet drop for multicast IPv4 or IPv6 as source IP is enabled or disabled for all traffic use the **show ip allow-src-multicast** command.

Configuring the maximum ICMP error message rate

NOTE
The maximum ICMP error message rate configuration only supports IPv4 traffic.

The Brocade device configuration allows 200 ICMP error messages per second per IP interface. You can now configure the maximum ICMP error message rate on all Interface Modules. The maximum configured value is increased to 5000 error messages per second. The maximum ICMP error message rate configuration uses an ICMP error metering mechanism. The process for the ICMP error metering mechanism is as follows:

- There is a meter counter for each interface. There is one total meter counter per Interface Module.
- The interface counter and the total counter will increment every time an icmp error message is sent out.
- The timer will rest all counters to 0 every second.
- Before an error message is sent out, it check the interface meter counter against the user configured icmp error limit (5000 max). The total counter will check against 10000. The error message is dropped if one any counter is larger the checked value.

The total error rate for all IP interfaces on an Interface Module is 10,000 errors per second. The ICMP error metering mechanism is per IP interface; this includes VRF IP interfaces.

Since the ICMP error metering code implementation is similar between the Management Module and Interface Module code, this change will also affect the Management Module ICMP error rate.

To configure the maximum ICMP error rate, enter the following command.

```
device(config)# ip icmp max-err-msg-rate 600
```

Syntax: [no] ip icmp max-err-msg-rate error per second

The *error per second* variable specifies the maximum error rate in errors per second. The maximum configured value has a range from 0 (minimum) to 5000 (maximum) error message per second. The default value is 400.

Disabling ICMP messages

The Brocade device is enabled to reply to ICMP echo messages and send ICMP Destination Unreachable messages by default.

You can selectively disable the following types of Internet Control Message Protocol (ICMP) messages:

- **Echo messages (ping messages)** - The Brocade device replies to IP pings from other IP devices.
- **Destination Unreachable messages** - If the Brocade device receives an IP packet that it cannot deliver to its destination, the Brocade device discards the packet and sends a message back to the device that sent the packet. The message informs the device that the destination cannot be reached by the Brocade device.

Disabling replies to broadcast ping requests

By default, the Brocade device is enabled to respond to broadcast ICMP echo packets, which are ping requests.

To disable response to broadcast ICMP echo packets (ping requests), enter the following command.

```
device(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.

```
device(config)# ip icmp echo broadcast-request
```

Disabling ICMP destination unreachable messages

By default, when the Brocade device receives an IP packet that the device cannot deliver, the device sends an ICMP Unreachable message back to the host that sent the packet. You can selectively disable a Brocade device's response to the following types of ICMP Unreachable messages:

- **Administration** - The packet was dropped by the device due to a filter or ACL configured on the device.
- **Fragmentation-needed** - The packet has the Do not Fragment bit set in the IP Flag field, but the Brocade device cannot forward the packet without fragmenting it.
- **Host** - The destination network or subnet of the packet is directly connected to the Brocade device, but the host specified in the destination IP address of the packet is not on the network.
- **Network** - The Brocade device cannot reach the network specified in the destination IP address of the packet.
- **Port** - The destination host does not have the destination TCP or UDP port specified in the packet. In this case, the host sends the ICMP Port Unreachable message to the Brocade device, which in turn sends the message to the host that sent the packet.
- **Protocol** - The TCP or UDP protocol on the destination host is not running. This message is different from the Port Unreachable message, which indicates that the protocol is running on the host but the requested protocol port is unavailable.
- **Source-route-failure** - The device received a source-routed packet but cannot locate the next-hop IP address indicated in the packet's Source-Route option.

You can disable the Brocade device from sending these types of ICMP messages on an individual basis.

> NOTE
> Disabling an ICMP Unreachable message type does not change the Brocade device's ability to forward packets. Disabling ICMP Unreachable messages prevents the device from generating or forwarding the Unreachable messages.

To disable all ICMP Unreachable messages, enter the following command.

```
device(config)# no ip icmp unreachable
```

Syntax: [no] ip icmp unreachable [**network** | **host** | **protocol** | **administration** | **fragmentation-needed** | **port** | **source-route-fail**]

- If you enter the command without specifying a message type (as in the example above), all types of ICMP Unreachable messages listed above are disabled. If you want to disable only specific types of ICMP Unreachable messages, you can specify the message type. To disable more than one type of ICMP message, enter the **no ip icmp unreachable** command for each messages type.
- The **network** parameter disables ICMP Network Unreachable messages.
- The **host** parameter disables ICMP Host Unreachable messages.
- The **protocol** parameter disables ICMP Protocol Unreachable messages.
- The **administration** parameter disables ICMP Unreachable (caused by Administration action) messages.
- The **fragmentation-needed** parameter disables ICMP Fragmentation-Needed But Do not-Fragment Bit Set messages.
- The **port** parameter disables ICMP Port Unreachable messages.
- The **source-route-fail** parameter disables ICMP Unreachable (caused by Source-Route-Failure) messages.

To disable ICMP Host Unreachable messages and ICMP Network Unreachable messages but leave the other types of ICMP Unreachable messages enabled, enter the following commands instead of the command shown above.

```
device(config)# no ip icmp unreachable host
device(config)# no ip icmp unreachable network
```

If you have disabled all ICMP Unreachable message types but want to re-enable certain types, you can do so by entering commands such as the following.

```
device(config)# ip icmp unreachable host
device(config)# ip icmp unreachable network
```

These commands re-enable ICMP Unreachable Host messages and ICMP Network Unreachable messages.

Disabling ICMP redirect messages

ICMP redirect messages can be disabled or re-enabled. By default, the Brocade device sends an ICMP redirect message to the source of a misdirected packet in addition to forwarding the packet to the appropriate router. You can disable ICMP redirect messages on a global basis or on an individual port basis.

> NOTE
> An unusually high receipt of multiple Internet Control Message Protocol (ICMP) Redirect packets that are used to change routing table entries in a short period of time may cause high CPU utilization. This can be avoided by configuring the maximum ICMP error message rate using **ip icmp max-err-msg-rate** command, 0 (minimum) to 5000 (maximum) error message per second. The default value is 400.The total error rate for all IP interfaces (SYSTEM) is 10,000 errors per second.

> NOTE
> The device forwards misdirected traffic to the appropriate router, even if you disable the redirect messages.

To disable ICMP redirect messages globally, enter the following command at the global CONFIG level of the CLI.

> NOTE
> The **ip icmp redirects** command is applicable to the Brocade MLX Series and NetIron XMR Series devices only.

```
device(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.

```
device(config)# int e 3/11
device(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 Brocade device automatically creates a route for the network the interface is in.
- **RIP** - If RIP is enabled, the Brocade device can learn about routes from the advertisements other RIP routers send to the Brocade device. If the route has a lower administrative distance than any other routes from different sources to the same destination, the Brocade device places the route in the IP route table.
- **OSPF** - Refer to RIP, but substitute "OSPF" for "RIP".
- **BGP4** - Refer to RIP, but substitute "BGP4" for "RIP".
- **Default network route** - A statically configured default route that the Brocade device uses if other default routes to the destination are not available. Refer to Configuring a default network route on page 760.

- **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 Brocade device interface through which you want the Brocade device to send traffic for the route. Typically, this type of static route is for directly attached destination networks.
- **Null** - the static route consists of the destination network address and network mask, and the "null0" parameter. Typically, the null route is configured as a backup route for discarding traffic if the primary route is unavailable.

Static IP route parameters

When you configure a static IP route, you must specify the following parameters:

- The IP address and network mask for the route's destination network.
- The route's path, which can be one of the following:
  - The IP address of a next-hop gateway
  - An Ethernet port
  - A virtual interface (a routing interface used by VLANs for routing Layer 3 protocol traffic among one another)
  - A "null" interface. The Brocade device drops traffic forwarded to the null interface.

The following parameters are optional:

- **The route's metric** - The value the Brocade device uses when comparing this route to other routes in the IP route table to the same destination. The metric applies only to routes that the Brocade device has already placed in the IP route table. The default metric for static IP routes is 1.
- **The route's administrative distance** - The value that the Brocade device uses to compare this route with routes from other route sources to the same destination before placing a route in the IP route table. This parameter does not apply to routes that are already in the IP route table. The default administrative distance for static IP routes is 1.

The default metric and administrative distance values ensure that the Brocade device always prefers static IP routes over routes from other sources to the same destination.

Multiple static routes to the same destination provide load sharing and redundancy

You can add multiple static routes for the same destination network to provide one or more of the following benefits:

- **IP load balancing** - When you add multiple IP static routes for the same destination to different next-hop gateways, and the routes each have the same metric and administrative distance, the Brocade device can load balance traffic to the routes' destination. For information about IP load balancing, refer to Configuring IP load sharing on page 763.
- **Path redundancy** - When you add multiple static IP routes for the same destination, but give the routes different metrics or administrative distances, the Brocade device uses the route with the lowest administrative distance by default, but uses another route to the same destination of the first route becomes unavailable.

Refer to the following sections for examples and configuration information:

- Configuring load balancing and redundancy using multiple static routes to the same destination on page 752

- Configuring standard static IP routes and interface or null static routes to the same destination on page 753

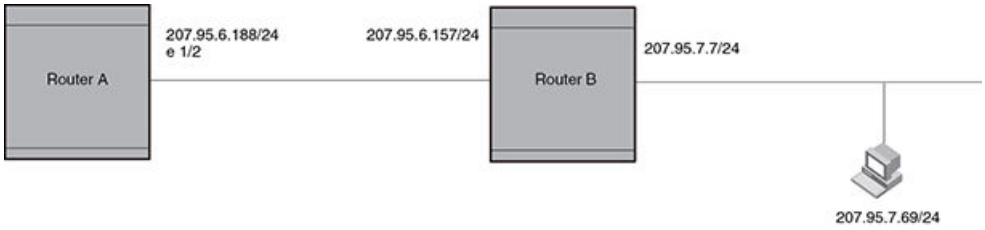
Static route states follow port states

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 Brocade device to adjust to changes in network topology. The Brocade device does not continue trying to use routes on unavailable paths but instead uses routes only when their paths are available.

Figure 195 shows a network containing a static route. The static route is configured on Router A, as shown in the CLI following the figure.

FIGURE 192 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.

```
device(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 Brocade device interface through which the Brocade device can reach the route. The Brocade device adds the route to the IP route table. In this case, Router A knows that 207.95.6.157 is reachable through port 1/2, and also assumes that local interfaces within that subnet are on the same port. Router A deduces that IP interface 207.95.7.188 is also on port 1/2.

The software automatically removes a static IP route from the IP route table if the port used by that route becomes unavailable. When the port becomes available again, the software automatically re-adds the route to the IP route table.

Configuring a static IP route

To configure an IP static route with a destination address of 10.0.0.0 255.0.0.0 and a next-hop router IP address of 10.1.1.1, enter the following.

```
device(config)# ip route 10.0.0.0 255.0.0.0 195.1.1.1
```

To configure a default route, enter the following.

```
device(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.

```
device(config)# ip route 10.128.2.69 255.255.255.0 ethernet 4/1
```

The command configures a static IP route for destination network 10.128.2.69/24. Since an Ethernet port is specified instead of a gateway IP address as the next hop, the Brocade device always forwards traffic for the 10.128.2.69/24 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.

```
device(config)# ip route 10.128.2.71 255.255.255.0 ve 3
```

Syntax: [no] ip route dest-ip-addr dest-mask | dest-ip-addr/mask-bitsnext-hop-ip-addr | ethernet slot/port | ve num [metric] [tag num] [distance num] [name string]

> NOTE
> Using the **no** command will only remove the name if configured. Run the **no** command again without the **name** parameter to remove the actual Static Route.

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 10.0.0.0 255.255.255.0 as 10.0.0.0/.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 Brocade device. The *num* parameter is a virtual interface number. The *slot/port* is the port's number of the Brocade device. If you specify an Ethernet port, the Brocade device forwards packets destined for the static route's destination network to the specified interface. Conceptually, this feature makes the destination network like a directly connected network, associated with a Brocade device interface.

> NOTE
> The port or virtual interface you use for the static route's next hop must have at least one IP address configured on it. The address does not need to be in the same subnet as the destination network.

The *metric* parameter specifies the cost of the route and can be a number from 1 - 16. The default is 1.

> NOTE
> If you specify 16, RIP considers the metric to be infinite and thus also considers the route to be unreachable.

The **tag** *num* parameter specifies the tag value of the route. Possible values: 0 - 4294967295. Default: 0.

The **distance***num* parameter specifies the administrative distance of the route. When comparing otherwise equal routes to a destination, the Brocade device prefers lower administrative distances over higher ones, so make sure you use a low value for your default route. Possible values: 1 - 255. Default: 1.

> NOTE
> The Brocade device will replace the static route if it receives a route with a lower administrative distance.

The **name***string* parameter specifies the name assigned to a route. The static route name is descriptive and an optional feature. It does not affect the selection of static routes.

> NOTE
> Using the **no ip route** command will only remove the name if configured. Run the **no** command again without the **name** parameter to remove the actual Static Route.

Configuring a static IP route between VRFs

You can configure a static route next hop to be in a different VRF. This can be done for the following:

- From the default VRF to a non-default VRF
- From a non-default VRF to a non-default VRF
- From a non-default VRF to the default VRF

- From one VRF to an IP interface in a different VRF.

  NOTE
  RPF is not supported with the Static Route between VRFs feature.

  NOTE
  For information on disabling gratuitous ARP requests on a VRF IP interface, refer to Disabling gratuitous ARP requests for local proxy ARP on page 716.

Configuring a static route from the default VRF to a non-default VRF

To configure an IP static route with a destination address of 10.0.0.0/24 and a next-hop router with an IP address of 10.1.1.1 in the non-default VRF named "blue", enter the following at the general configuration prompt.

```
device(config)# vrf red
device(config-vrf-red)# ip route 10.128.2.69/24 next-hop-vrf blue 10.1.1.1
```

Syntax: [no] ip route dest-ip-addr dest-mask | dest-ip-addr/mask-bitsnext-hop-vrf next-hop-vrf-name next-hop-ip-addr

The *dest-ip-addr* is the route's destination. The *dest-mask* is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 10.0.0.0 255.255.255.0 as 192.0.0.0/.24.

The *next-hop-vrf-name* is the name of the VRF that contains the next-hop router (gateway) for the route.

The *next-hop-ip-addr* is the IP address of the next-hop router (gateway) for the route.

NOTE
The **next-hop-vrf** needs to be a valid VRF to be used in this command.

Configuring a static route from a non-default VRF to a non-default VRF

To configure an IP static route within the VRF named "red" with a destination address of 10.2.2.0/24 and a next-hop router with an IP address of 10.2.2.1 in the non-default VRF named "blue", enter the following commands from within the VRF "red" configuration context.

For the VRF configuration you need to configure a Route Descriptor (RD) and address-family IPv4 before you can enter the **ip route** command.

```
device(config)# vrf red
device(config-vrf-red)# rd 3:1
device(config-vrf-red)# address-family ipv4
device(config-vrf-red)# ip route 10.128.2.69/24 next-hop-vrf blue 10.1.1.1
```

Syntax: [no] ip route dest-ip-addr dest-mask | dest-ip-addr/mask-bitsnext-hop-vrf next-hop-vrf-name next-hop-ip-addr

The *dest-ip-addr* is the route's destination. The *dest-mask* is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 10.0.0.0 255.255.255.0 as 10.0.0.0/.24.

The *next-hop-vrf-name* is the name of the VRF that contains the next-hop router (gateway) for the route.

The *next-hop-ip-addr* is the IP address of the next-hop router (gateway) for the route.

NOTE
The **next-hop-vrf** needs to be a valid VRF to be used in this command.

Configuring a static route from a non-default VRF to the default VRF

To configure an IP static route within the VRF named "red" with a destination address of 10.0.0.0/24 and a next-hop router in the default VRF and an IP address of 10.1.1.1, enter the following from within the VRF "red" configuration context.

For the VRF configuration you need to configure a Route Descriptor (RD) and address-family IPv4 before you can enter the **ip route** command.

```
device(config)# vrf red
device(config-vrf-red)# rd 3:1
device(config-vrf-red)# address-family ipv4
device(config-vrf-red)# ip route 10.128.2.69/24 next-hop-vrf default-vrf 10.1.1.1
```

Syntax: [no] ip route dest-ip-addr dest-mask | dest-ip-addr/mask-bitsnext-hop-vrf next-hop-vrf-name next-hop-ip-addr

The *dest-ip-addr* is the route's destination. The *dest-mask* is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering / followed by the number of bits in the network mask. For example, you can enter 10.0.0.0 255.255.255.0 as 10.0.0.0/.24.

The **default-vrf** option specifies that the next-hop router (gateway) for the route is in the default VRF.

The *next-hop-ip-addr* is the IP address of the next-hop router (gateway) for the route.

> NOTE
> The **next-hop-vrf** needs to be a valid VRF to be used in this command.

Configuring an IP static interface route across VRFs

You can configure an IP Static interface route from one VRF to an IP interface in a different VRF. This allows you to connect from one VRF to a host that is directly connected to a port in a different VRF. You can do this by configuring a static route to point to the interface that is directly connected to the device with the IP address you want to reach. The following example defines two VRFs as follows:

VRF A :

Route Distinguisher = 1:1

Interface: ethernet port 1/1

IP address: 10.0.0.1/24

VRF B :

Route Distinguisher = 2:2

Interface: ethernet port 1/2

IP address: 10.0.0.1/24

```
device(config)# vrf A
device(config-vrf-A)# rd 1:1
device(config-vrf-A)# exit
device(config)# interface ethernet 1/1
device(config-if-e10000-1/1)# vrf forwarding A
device(config-if-e10000-1/1)# ip address 10.0.0.1/24
device(config-if-e10000-1/1)# exit
device(config)# vrf B
device(config-vrf-B)# rd 2:2
device(config-vrf-B)# exit
device(config)# interface ethernet 1/2
device(config-if-e10000-1/2)# vrf forwarding B
device(config-if-e10000-1/2)# ip address 10.0.0.1/24
```

The following example configures an IP Static interface route from VRF A to a network with IP address 10.0.0.0/24, which is directly connected to ethernet port 1/2 in VRF B.

For the VRF configuration you need to configure a Route Descriptor (RD) and address-family IPv4 before you can enter the **ip route** command.

```
device(config)# vrf a
device(config-vrf-A)# rd 1:1
device(config-vrf-A)# address-family ipv4
device(config-vrf-A)# ip route 10.0.0.0/24 ethernet 1/2
```

Syntax: [no] ip route dest-ip-addr/mask-bits [**ethernet slot/port** | **ve num**]

The *dest-ip-addr* is the route's destination. The *dest-mask* is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 10.0.0.0 255.255.255.0 as 10.0.0.0/.24. To configure a default route, enter 0.0.0.0 for *dest-ip-addr* and 0.0.0.0 for *dest-mask* (or 0 for the *mask-bits* if you specify the address in CIDR format). Specify the IP address of the default gateway using the *next-hop-ipaddr* parameter.

The *slot/port* or *num* is an interface in a different VRF that is directly connected to the device that you want to reach.

Configuring a "null" route

You can configure the Brocade device to drop IP packets to a specific network or host address by configuring a "null" (sometimes called "null0") static route for the address. When the Brocade device receives a packet destined for the address, the Brocade device drops the packet instead of forwarding it.

To configure a null static route to drop packets destined for network 10.157.22.x, enter the following commands.

```
device(config)# ip route 10.157.22.0 255.255.255.0 null0
device(config)# write memory
```

Syntax: [no] ip route ip-addr ip-mask | **dest-ip-addr/mask-bits null0** [**metric**] [**tag num**] [**distance num**]

To display the maximum value for your device, enter the **show default values** command. The maximum number of static IP routes the system can hold is listed in the ip-static-route row in the System Parameters section of the display. To change the maximum value, use the **system-max ip-static-route** command at the global CONFIG level.

The *ip-addr* parameter specifies the network or host address. The Brocade device will drop packets that contain this address in the destination field instead of forwarding them.

The *ip-mask* parameter specifies the network mask. Ones are significant bits and zeros allow any value. For example, the mask 255.255.255.0 matches on all hosts within the Class C subnet address specified by *ip-addr*. Alternatively, you can specify the number of bits in the network mask. For example, you can enter 10.157.22.0/24 instead of 10.157.22.0 255.255.255.0.

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 Brocade device's CPU.

Hardware dropping for IP traffic sent to the null0 interface is supported.

You can optionally configure the Brocade device to drop traffic sent to the default IP route address in hardware. To do this, enter the following commands.

```
device(config)# ip route 0.0.0.0 0.0.0.0 null0
device(config)# ip hw-drop-on-def-route
```

Syntax: [no] ip hw-drop-on-def-route

CAM default route aggregation

Configuring the Brocade device to drop traffic sent to the default IP route address in hardware causes the device to program 32-bit host CAM entries for each destination address using the default route, which could consume the CAM space. To prevent this from happening, you can enable the CAM Default Route Aggregation feature. To do this, enter the following command.

```
device(config)# ip dr-aggregate
```

Syntax: [no] ip dr-aggregate

Configuring load balancing and redundancy using multiple static routes to the same destination

You can configure multiple static IP routes to the same destination, for the following benefits:

- **IP load sharing** - If you configure more than one static route to the same destination, and the routes have different next-hop gateways but have the same metrics, the Brocade device load balances among the routes using basic round-robin. For example, if you configure two static routes with the same metrics but to different gateways, the Brocade device alternates between the two routes. For information about IP load balancing, refer to Configuring IP load sharing on page 763.
- **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 Brocade device will always use the route with the lowest metric. If this route becomes unavailable, the Brocade device will fail over to the static route with the next-lowest metric, and so on.

  NOTE
  You also can bias the Brocade device to select one of the routes by configuring them with different administrative distances. However, make sure you do not give a static route a higher administrative distance than other types of routes, unless you want those other types to be preferred over the static route.

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.

```
device(config)# ip route 10.128.2.69 255.255.255.0 10.157.22.1
device(config)# ip route 10.128.2.69 255.255.255.0 10.111.10.1
```

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 Brocade device uses the route with the lowest metric if the route is available.

```
device(config)# ip route 10.128.2.69 255.255.255.0 10.157.22.1
device(config)# ip route 10.128.2.69 255.255.255.0 10.111.10.1 2
device(config)# ip route 10.128.2.69 255.255.255.0 10.1.1.1 3
```

In this example, each static route has a different metric. The metric is not specified for the first route, so the default (1) is used. A metric is specified for the second and third static IP routes. The second route has a metric of two and the third route has a metric of 3. Thus, the second route is used only of the first route (which has a metric of 1) becomes unavailable. Likewise, the third route is used only if the first and second routes (which have lower metrics) are both unavailable.

For complete syntax information, refer to Configuring a static IP route on page 747.

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 Brocade device has multiple routes to the same destination, the Brocade device always prefers the route with the lowest metric. Generally, when you configure a static route to a destination network, you assign the route a low metric so that the Brocade device prefers the static route over other routes to the destination.

This feature is especially useful for the following configurations. These are not the only allowed configurations but they are typical uses of this enhancement:

- When you want to ensure that if a given destination network is unavailable, the Brocade device drops (forwards to the null interface) traffic for that network instead of using alternate paths to route the traffic. In this case, assign the normal static route to the destination network a lower metric than the null route.
- When you want to use a specific interface by default to route traffic to a given destination network, but want to allow the Brocade device to use other interfaces to reach the destination network if the path that uses the default interface becomes unavailable. In this case, give the interface route a lower metric than the normal static route.

  NOTE
  You cannot add a null or interface-based static route to a network if there is already a static route of any type with the same metric you specify for the null or interface-based route.

Figure 196 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 Brocade device always prefers the static route with the lower metric. In this example, the Brocade device always uses the standard static route for traffic to destination network 192.168.7.0/24, unless that route becomes unavailable, in which case the Brocade device sends traffic to the null route instead.

FIGURE 193 Standard and null static routes to the same destination network

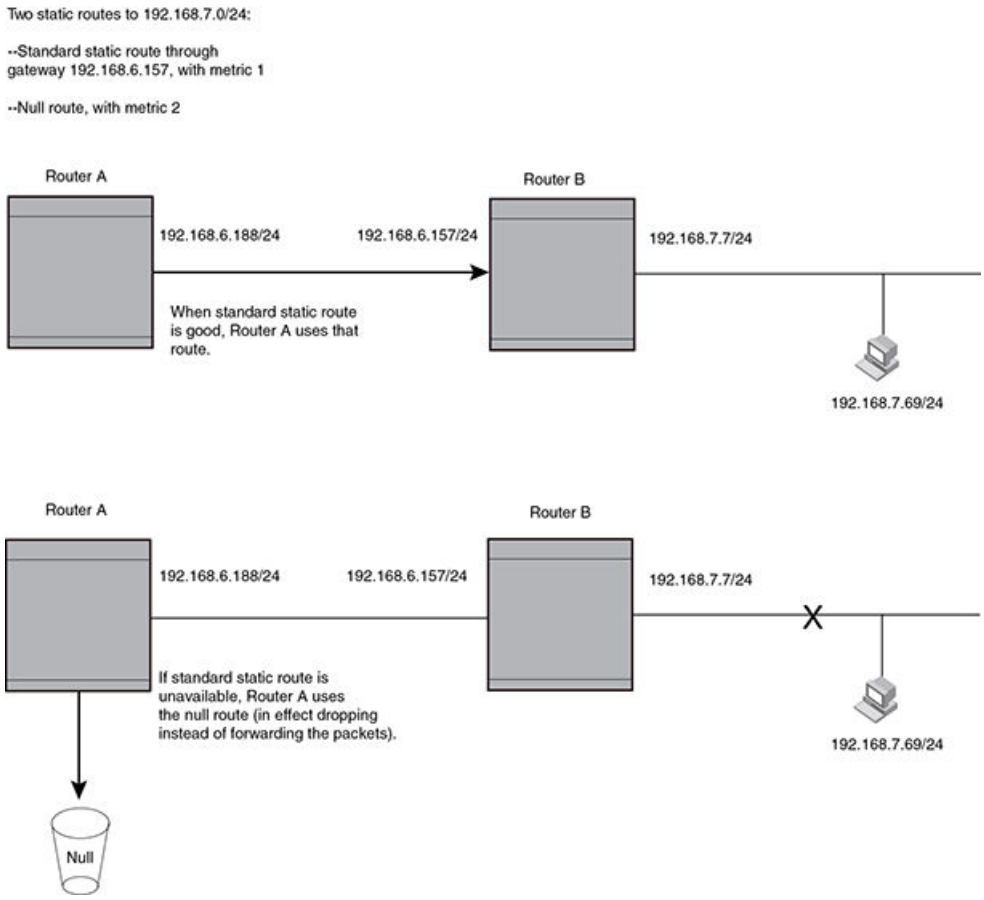
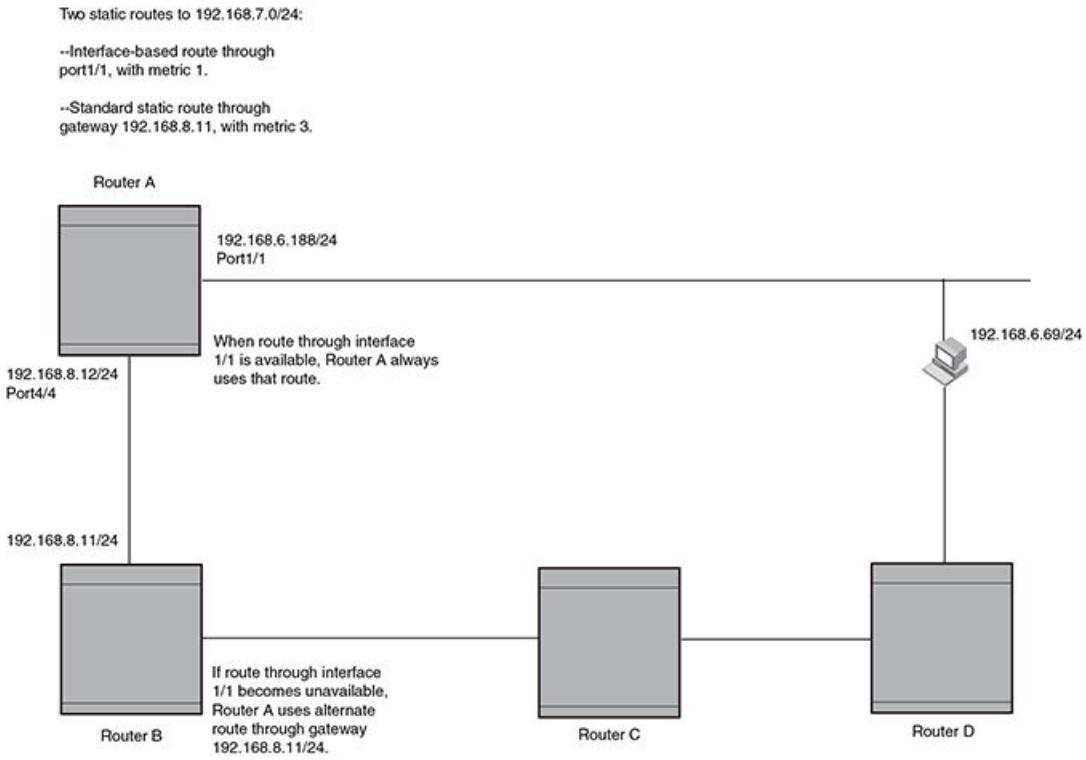


Figure 197 shows another example of two static routes. A standard static route and an interface-based static route are configured for destination network 192.168.6.0/24. The interface-based static route has a lower metric than the standard static route. As a result, the Brocade device always prefers the interface-based route when the route is available. However, if the interface-based route becomes unavailable, the Brocade device still forwards the traffic toward the destination using an alternate route through gateway 192.168.8.11/24.

FIGURE 194 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 196, enter commands such as the following.

```
device(config)# ip route 192.168.7.0/24 192.168.6.157/24 1
device(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 Brocade device to always prefer this route when the route is available.

The second command configures another static route for the same destination network, but the second route is a null route. The metric for the null route is 3, which is higher than the metric for the standard static route. If the standard static route is unavailable, the software uses the null route.

For complete syntax information, refer to Configuring a static IP route on page 747.

To configure a standard static route and an interface-based route to the same destination, enter commands such as the following.

```
device(config)# ip route 192.168.6.0/24 ethernet 1/1 1
device(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 Brocade device to always prefer this route when it is available. If the route becomes unavailable, the Brocade device uses an alternate route through the next-hop gateway 192.168.8.11/24.

Static route configuration

The following enhancements to static route configuration have been added:

Static route tagging

Static routes can be configured with a tag value, which can be used to color routes and filter routes during a redistribution process. When tagged static routes are redistributed to OSPF or to a protocol that can carry tag information, they are redistributed with their tag values.

To add a tag value to a static route, enter commands such as the following.

```
device(config)# ip route 10.122.12.1 255.255.255.0 10.122.1.1 tag 20
```

Syntax: [no] ip route dest-ip-addr dest-mask | dest-ip-addr/dest-mask next-hop-ip-address tag value

The *dest-ip-addr* is the route's destination. The *dest-mask* is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 10.0.0.0 255.255.255.0 as 10.0.0.0/.24. You can enter multiple static routes for the same destination for load balancing or redundancy.

The *next-hop-ip-address* is the IP address of the next-hop router (gateway) for the route. In addition, the *next-hop-ip-address* can also be a virtual routing interface (for example. ve 100), or a physical port (for example, ethernet 1/1) that is connected to the next-hop router.

Enter 0 - 4294967295 for **tag***value*. The default is 0, meaning no tag.

Static route next hop resolution

This feature of the Multi-Service IronWare software enables the Brocade device to use routes from a specified protocol to resolve a configured static route. By default this is disabled.

To configure static route next hop resolution with OSPF routes, use the following command.

```
device(config)# ip route next-hop ospf
```

Syntax: [no] ip route next-hop [**bgp** | **isis** | **ospf** | **rip**]

NOTE
This command can be independently applied on a per-VRF basis.

This command causes the resolution of static route next hop using routes learned from one of the following protocols:

- bgp - both iBGP and eBGP routes are used to resolve static routes.
- isis
- ospf
- rip

NOTE
Connected routes are always used to resolve static routes.

Static route recursive lookup

This feature of the Multi-Service IronWare software enables the Brocade device to use static routes to resolve another static route. The recursive static route next hop lookup level can be configured. By default, this feature is disabled.

To configure static route next hop recursive lookup by other static routes, use the following command.

```
device(config)# ip route next-hop-recursion 5
```

Syntax: [no] ip route next-hop-recursion level

The *level* available specifies the numbers of level of recursion allowed. Acceptable values are 1-10. This feature is disabled by default. When enabled, the default value is 3.

NOTE
This command can be independently applied on a per-VRF basis.

Static route resolve by default route

This feature of the Multi-Service IronWare software enables the Brocade device to use the default route (0.0.0.0/0) to resolve a static route. By default, this feature is disabled.

Use the following command to configure static route resolve by default route.

```
device(config)# ip route next-hop-enable-default
```

Syntax: [no] ip route next-hop-enable-default

NOTE
This command can be independently applied on a per-VRF basis.

NOTE
This command works independently with the **ip route next-hop-recursion** and **ip route next-hop** commands. If the default route is a protocol route, that protocol needs to be enabled to resolve static routes using the **ip route next-hop** command with **protocol-name parameter** in order for static routes to resolve by this default route. If the default route itself is a static route, you must configure the **ip route next-hop-recursion** command to resolve other static routes by this default route.

Static route to an LSP tunnel interface

This feature allows you to set the next hop for a static route to the egress router of an LSP tunnel if the destination route is contained in the MPLS routing table. In this configuration, the static route is updated with the LSP routes and reverts to its original next hop outgoing interface when this feature is disabled or when the LSP goes down. This route can be used for the default route.

To enable the static route to an LSP tunnel interface feature, use the following command.

```
device(config)# ip route next-hop-enable-mpls
```

Syntax: [no] ip route next-hop-enable-mpls

The static route can then be directed to the IP address of the egress router of the LSP. In the following example, a static route is configured to network 10.10.10.0/24 through 10.11.11.1, which is the IP address of the egress router of an LSP tunnel.

```
device(config)# ip route 10.10.10.0/24 10.11.11.1
```

As previously stated, this feature works only if a route to the destination network is contained in the MPLS routing table. To verify that it is, you can use the **show ip route** command, as shown in the following example.

```
device# show ip route
Total number of IP routes: 6
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
        Destination        Gateway         Port           Cost          Type
1       10.47.6.0/24       DIRECT          mgmt 1         0/0           D
2       10.11.11.11/32     DIRECT          loopback 1     0/0           D
3       10.12.12.12/32     10.1.0.1        eth 5/1        110/3         O
```

```
          10.12.12.12/32     10.12.12.12     lsp brocadel   110/3          O
                                             onglspfoundr
                                             ylonglspfoun
                                             drylonglspfo
                                             undrylonglsp
                                             fou
          10.12.12.12/32     10.12.12.12     lsp t11        110/3          O
4         10.13.13.13/32     10.1.0.1        eth 5/1        110/2          O
5         10.1.0.0/24        DIRECT          eth 5/1        0/0            D
6         10.1.0.0/24        10.1.0.1        eth 5/1        110/2          O
device#
```

As shown in the example, route 2 has a destination network of 10.10.10.0/24 through the gateway at IP address 10.11.11.1 which is on LSP 12.

NOTE

The show commands are enhanced to include the full LSP name. Previously, the LSP name was truncated because it exceeded the character length. Now, the LSP name is text wrapped to display the full name. For example, see show ip route above.

To verify that an LSP is up and that MPLS has a route to it, you can use the **show mpls lsp** and **show mpls route** commands as shown in the following examples.

```
device# show mpls lsp
Note: LSPs marked with * are taking a Secondary Path
                              Admin  Oper  Tunnel   Up/Dn  Retry   Active
Name           To             State State Intf     Times  No.     Path
t6             10.12.12.13    UP    DOWN    --      0      292     --
t5             10.12.12.12    UP    UP     tnl3     1      0       --
t2             10.12.12.12    UP    UP     tnl1     1      0       --
t7             10.12.12.12    UP    UP     tnl5     1      0       one
brocadelongl   10.12.12.12    UP    UP     tnl9     1      0       --
spverylongls
pveryverylon
glsp
t4             10.12.12.12    UP    UP     tnl2     1      0        --
t1             10.12.12.12    UP    UP     tnl0     1      0         --

device(config-mpls-if-e100-1/3)#show mpls route
Total number of MPLS tunnel routes: 3
R:RSVP L:LDP S:Static O:Others
      Destination          Gateway          Tnnl    Port  Label  Sig Cost Use
1     10.2.2.2/32           10.2.2.2          tnl8    1/2   3      L   0    0
2     10.3.3.3/32           10.2.2.2          tnl9    1/1   1025   L   0    0
3     10.3.3.3/32           10.3.3.3          tnl1    1/2   1027   R   0    0
```

NOTE

The show commands have been enhanced to include the full MPLS tunnel name. Previously, the MPLS tunnel name was truncated because it exceeded the character length. Now, the MPLS tunnel name is text-wrapped to display the full name. For an illustration, see the output of the **show mpls lsp** and **show mpls route** commands in the preceding examples.

Naming a static IP route

You can assign a name to a static IP route. A static IP route name serves as a description of the route. The name can be used to more readily reference or identify the associated static route.

NOTE

The static route name is an optional feature. It does not affect the selection of static routes.

The Brocade device does not check for the uniqueness of names assigned to static routes. Static routes that have the same or different next hop(s) can have the same or different name(s). Due to this, the same name can be assigned to multiple static routes to group them. The name is then used to reference or identify a group of static routes.

NOTE
This feature is supported on standard static IP routes and static IP routes between VRFs (both default and non-default).

The option to assign a name to a static route is displayed after you select either an outgoing interface type or configure the next hop address.

To assign a name to a static route, enter commands such as the following.

```
device(config)# ip route 10.22.22.22 255.255.255.255 eth 1/1 name abc
```

OR

```
device(config)# ip route 10.22.22.22 255.255.255.255 10.1.1.1 name abc
```

Syntax: [no] ip route *dest-ip-addrdest-mask* | *dest-ip-addr/mask-bitsnext-hop-ip-addr* | **ethernet** *slot/port* | **ve num** [**metric**] [**tag num**] [**distance num**] [**name** *string*]

Enter the static route name for **name***string*. The maximum length of the name is 128 bytes.

The output of the **show** commands displays the name of a static IP route if there is one assigned.

The **show run** command displays the entire name of the static IP route. The **show ip static route** command displays an asterisk (*) after the first twelve characters if the assigned name is thirteen characters or more. The **show ipv6 static route** command displays an asterisk after the first two characters if the assigned name is three characters or more.

When displayed in **show run** , a static route name with a space in the name will appear within quotation marks (for example, "brcd route").

Changing the name of a static IP route

To change the name of a static IP route, enter the static route as configured. Proceed to enter the new name instead of the previous name. See the example below.

Static IP route with the original name "abc":

```
device(config)# ip route 10.22.22.22 255.255.255.255 10.1.1.1 name abc
```

Change the name of "abc" to "xyz":

```
device(config)# ip route 10.22.22.22 255.255.255.255 10.1.1.1 name xyz
```

In this example, "xyz" is the set as the new name of the static IP route.

Deleting the name of a static IP route

To delete the name of a static IP route, use the **no** command. See the example below.

Static IP route with the name "xyz":

```
device(config)# ip route 10.22.22.22 255.255.255.255 10.1.1.1 name xyz
```

To remove the name "xyz" from the static IP route, specify both "name" and the string, in this case "xyz".

```
device(config)#no ip route 10.22.22.22 255.255.255.255 10.1.1.1 name xyz
```

The static route no longer has a name assigned to it.

Configuring a default network route

The Brocade device enables you to specify a candidate default route without the need to specify the next hop gateway. If the IP route table does not contain an explicit default route (for example, 0.0.0.0/0) or propagate an explicit default route through routing protocols, the software can use the default network route as a default route instead.

When the software uses the default network route, it also uses the default network route's next hop gateway as the gateway of last resort.

This feature is especially useful in environments where network topology changes can make the next hop gateway unreachable. This feature allows the Brocade device to perform default routing even if the default network route's default gateway changes.

The feature thus differs from standard default routes. When you configure a standard default route, you also specify the next hop gateway. If a topology change makes the gateway unreachable, the default route becomes unusable.

For example, if you configure 10.10.10.0/24 as a candidate default network route, if the IP route table does not contain an explicit default route (0.0.0.0/0), the software uses the default network route and automatically uses that route's next hop gateway as the default gateway. If a topology change occurs and as a result the default network route's next hop gateway changes, the software can still use the default network route.

If you configure more than one default network route, the Brocade device uses the following algorithm to select one of the routes.

1. Use the route with the lowest administrative distance.
2. If the administrative distances are equal:
   - – Are the routes from different routing protocols (RIP, OSPF, or BGP4)? If so, use the route with the lowest IP address.
   - – If the routes are from the same routing protocol, use the route with the best metric. The meaning of "best" metric depends on the routing protocol:
   - – **RIP** - The metric is the number of hops (additional routers) to the destination. The best route is the route with the fewest hops.
   - – **OSPF** - The metric is the path cost associated with the route. The path cost does not indicate the number of hops but is instead a numeric value associated with each route. The best route is the route with the lowest path cost.
   - – **BGP4** - The metric is the Multi-exit Discriminator (MED) associated with the route. The MED applies to routes that have multiple paths through the same AS. The best route is the route with the lowest MED.

     NOTE
     Currently the Brocade device will *not* propagate a candidate default route, specified by the **ip default-network** command, into the routing protocols in spite of the **default-information-originate** command being configured under the routing protocols.

Configuring a default network route

NOTE
The **ip default-network** command is not supported on the Brocade NetIron CES Series and Brocade NetIron CER Series devices.

You can configure up to four default network routes. To configure a default network route, enter commands such as the following.

```
device(config)# ip default-network 10.157.22.0
device(config)# write memory
```

Syntax: [no] ip default-network ip-addr

The *ip-addr* parameter specifies the network address.

To verify that the route is in the route table, enter the following command at any level of the CLI.

```
device(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     10.157.20.0      0.0.0.0          lb1    1      D
2     10.157.22.0      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.

BFD for static routes

Unlike dynamic routing protocols, such as OSPF and BGP, static routing has no method of peer discovery and fault detection. BFD for IPv4 and IPv6 static routes provides rapid detection of failure in the bidirectional forwarding path between BFD peers.

BFD for Static Routes allows you to detect failures that impact the forwarding path of a static route. This feature supports both singlehop and multihop BFD Static Routes for both IPv4 and IPv6. Unless the BFD session is up, the gateway for the static route is considered unreachable, and the affected routes are not installed in the routing table. BFD can remove the associated static route from the routing table if the next-hop becomes unreachable indicating that the BFD session has gone down.

Static routes and BFD neighbors are configured separately. A static route is automatically associated with a static BFD neighbor if the static route's next-hop exactly matches the neighbor address of the static BFD neighbor and BFD monitoring is enabled for the static route.

When a static BFD neighbor is configured, BFD asks the routing table manager (RTM) if there is a route to the neighbor. If a route exists, and if the route is directly connected, then BFD initiates a single hop session. If the route is not directly connected, BFD establishes a multi-hop session. Once the session comes up, BFD adds the corresponding static routes to RTM. If no route exists, then BFD will not add the corresponding static routes to RTM.

When a BFD session goes down because the BFD neighbor is no longer reachable, static routes monitored by BFD are removed from the routing table manager. The removed routes can be added back if the BFD neighbor becomes reachable again. Singlehop BFD sessions use the BFD timeout values configured on the outgoing interface. Timeout values for multihop BFD sessions are specified along with each BFD neighbor. Multiple static routes going to the same BFD neighbor use the same BFD session and timeout values.

Configuration considerations

- In a multi-hop session, the protocol must be stated in the **ip route next-hop** command.
- BFD multi-hop is supported for a nexthop resolved through OSPF, BGP, ISIS, RIP, and MPLS.
- BFD multi-hop is not supported for a nexthop resolved through Default Route.
- BFD for static routes is not supported for static routes with an LSP name as nexthop.
- Upon reboot, the router will first bring up the static BFD sessions, and then install the static routes in the routing table manager (RTM). There may be a delay of 90 seconds before the BFD sessions become available.
- BFD is not supported in MCT.
- BFD for static routes will not support interface-based static routes for both IPv4 and IPv6.
- When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

Configuring BFD for static routes

The following example assumes the configured interface Ethernet 1/1 is as follows:

```
interface ethernet 1/1
  bfd interval 100 min-rx 100 multiplier 5
  ip address 10.0.0.1/24
```

Single hop configuration

To configure BFD for static routes, configure BFD neighbors using the following commands. BFD neighbors can be configured in unassociated mode using this command.

The following example uses interface ethernet 1/1 as the outgoing interface and uses the BFD intervals on ethernet 1/1. The next hop address 10.0.0.5 is the BFD neighbor and the configured address 10.0.0.1 on Ethernet 1/1 is the local address.

```
device(config)#ip route static-bfd 10.0.0.5 10.0.0.1
```

Syntax to configure BFD Static neighbor for IPv4:

Syntax: [no] ip route [**vrf** *vrf-name*] **static-bfd** *neighbor-ip-address local-ip-address* **interval** *tx-rate min-rxrx-rate* **multiplier value**

Syntax to configure BFD Static neighbor for IPv6:

Syntax: [no] ipv6 route [**vrf** *vrf-name*] **static-bfd** *neighbor-ipv6-address local-ipv6-address* **interval** *tx-rate min-rx rx-rate* **multiplier** *value*

The **no** version of the command removes the BFD monitoring by removing the BFD static neighbor 10.0.0.5 and eliminating the BFD session, while keeping the static route in the RTM, and retaining the existing traffic to IP route 20.0.0.0. You only need to specify the BFD neighbor address and the local address when removing a BFD neighbor.

To enable BFD for static routes use the following command. The bfd parameter allows you to enable BFD monitoring for the static route.

```
device(config)#ip route 20.0.0.0/24 10.0.0.5 bfd
```

Syntax to enable BFD monitoring for IPv4:

Syntax: [no] ip route *Destination IPaddress Next hop Router IPaddress* **... bfd**

Syntax to enable BFD monitoring for IPv6:

Syntax: [no] ipv6 route *Destination IPv6address Next hop Router IPv6address* **... bfd**

The **no** version of the command removes BFD monitoring from the static route.

Multi-hop configuration

The following example shows a multi-hop configuration using the commands explained in the single hop section.

```
device(config)#ip route static-bfd 30.0.0.5 10.0.0.1 interval 90 min-rx 90 multiplier 3
device(config)#ip route 20.0.0.0/24 30.0.0.5 bfd
```

The **multi-hop BFD** session to the next hop (BFD neighbor) 30.0.0.5 uses the TX and RX intervals of 90ms.

When configuring **multi-hop static route** and **multi-hop bfd neighbor** , the protocol by which the nexthop is to be resolved must be stated using the IP route next-hop command.

Show commands

The **show ip static route** and **show ipv6 static route** command output indicates that BFD monitoring is enabled by the **b** next to the static route.

```
Brocade# show ip static route
IP Static Routing Table - 3 entries:
STATIC Codes - b:BFD monitoring
   IP Prefix          Next Hop        Interface  Dis/Metric/Tag  Name
*  0.0.0.0/0          10.37.73.129    -          1/1/0
   0.0.0.0/0          10.37.73.1      -          1/1/0
 b
 100.0.0.0/8        10.0.0.2        -          1/1/0
 b
150.0.0.0/8        20.0.0.3        -          1/1/0
Brocade#
Brocade# show ipv6 static route
IPv6 Static Routing Table - 2 entries:
STATIC Codes - b:BFD monitoring
   IPv6 Prefix        Interface  Next Hop Router      Met/Dis/Tag Name
 b
100::/64            eth 1/5    10::2               1/1/0
 b
150::/64            eth 1/5    20::3               1/1/0
Brocade#
```

The **show bfd applications** output indicates that BFD monitoring is enabled by the **static** and **static6** .

```
Brocade# show bfd applications
  Registered Protocols Count: 4
     Protocol  VRFID     Parameter HoldoverInterval
     static6
   0         1         0
     static
    0         1         0
     bgp       1         0         0
     ospf      0         0         0
Brocade#
```

The **show bfd neighbors details** output indicates that BFD monitoring is enabled by the **static** and **static6** .

```
Brocade# show bfd neighbors details 20.0.0.3
NeighborAddress        State  Interface Holddown   Interval  R/H
20.0.0.3               UP     eth 1/5   300000     100000    Y/M
  Registered Protocols(Protocol/VRFID): static/0
  Local: Disc: 3, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Remote: Disc: 5, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Stats: RX: 89596 TX: 87853 SessionUpCount: 1 at SysUpTime: 0:5:10:53.575
  Session Uptime: 0:1:0:16.300, LastSessionDownTimestamp: 0:0:0:0.0
  Tx Port: eth 1/1(eth 1/1),Rx Port: eth 1/1(eth 1/1)
  Using PBIF Assist: Y
Brocade#
```

Configuring IP load sharing

The IP route table can contain more than one path to a given destination. When this occurs, the Brocade device selects the path with the lowest cost 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 Brocade device uses IP load sharing to select a path to the destination.

IP load sharing is based on the destination address of the traffic. Brocade devices support load sharing based on individual host addresses or on network addresses.

You can enable a Brocade device to load balance across up to eight equal-cost paths. The default maximum number of equal-cost load sharing paths is four.

NOTE
IP load sharing is not based on source routing, only on next-hop routing.

NOTE
The term "path" refers to the next-hop router to a destination, not to the entire route to a destination. Thus, when the software compares multiple equal-cost paths, the software is comparing paths that use different next-hop routers, with equal costs, to the same destination.In many contexts, the terms "route" and "path" mean the same thing. Most of the user documentation uses the term "route" throughout. The term "path" is used in this section to refer to an individual next-hop router to a destination, while the term "route" refers collectively to the multiple paths to the destination. Load sharing applies when the IP route table contains multiple, equal-cost paths to a destination.

NOTE
The Brocade device also performs load sharing among the ports in aggregate links.

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 Brocade device is configured with a unique administrative distance value for each IP route source.

When the software receives paths from different sources to the same destination, the software compares their administrative distances, selects the one with the lowest distance, and puts it in the IP route table. For example, if the Brocade device has a path learned from OSPF and a path learned from RIP for a given destination, only the path with the lower administrative distance enters the IP route table.

Here are the default administrative distances on the Brocade device:

- Directly connected - 0 (this value is not configurable)
- Static IP route - 1 (applies to all static routes, including default routes and default network routes)
- Exterior Border Gateway Protocol (EBGP) - 20
- OSPF - 110
- RIP - 120
- Interior Gateway Protocol (IBGP) - 200
- Local BGP - 200
- Unknown - 255 (the device will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the device receives routes for the same network from OSPF and from RIP, the device will prefer the OSPF route by default.

NOTE
You can change the administrative distances individually. Refer to the configuration chapter for the route source for information.

Since the software selects only the path with the lowest administrative distance, and the administrative distance is determined by the path's source, IP load sharing does not apply to paths from different route sources. IP load sharing applies only when the IP route table contains paths from the same IP route source to the same destination.

Path cost

The cost parameter provides a basis of comparison for selecting among paths to a given destination. Each path in the IP route table has a cost. When the IP route table contains multiple paths to a destination, the Brocade device chooses the path with the lowest cost. When the IP route table contains more than one path with the lowest cost to a destination, the Brocade device uses IP load sharing to select one of the lowest-cost paths.

The source of a path's cost value depends on the source of the path:

- **IP static route** - The value you assign to the metric parameter when you configure the route. The default metric is 1. Refer to Configuring load balancing and redundancy using multiple static routes to the same destination on page 752.
- **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 87 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 Brocade device, load sharing for static IP routes, RIP routes, OSPF routes, and BGP4 routes also is enabled by default.

TABLE 87 Default load sharing parameters for route sources

Route source	Default maximum number of paths	Maximum number of paths
Static IP route	4 NOTE This value depends on the value for IP load sharing, and is not separately configurable.	32 NOTE This value depends on the value for IP load sharing, and is not separately configurable.
RIP	4	8

TABLE 87 Default load sharing parameters for route sources (continued)

Route source	Default maximum number of paths	Maximum number of paths
	NOTE This value depends on the value for IP load sharing, and is not separately configurable.	NOTE This value depends on the value for IP load sharing, and is not separately configurable.
OSPF	4	32
BGP4	1	32

NOTE

Suppose you have a route that points to an ECMP next hop and the route paths consist of more than one type, then only the first path is programmed in the hardware for forwarding. The number of paths for ECMP is set to 1.

Options for IP load sharing and LAGs

The following options have been added to refine the hash calculations used for IP load sharing and LAGs. These include the following:

- **Speculate UDP or TCP Headers** - This option is applied to ECMP and LAG index hash calculations.
- **Mask Layer-3 and Layer-4 Information** - This option is applied to ECMP and LAG index hash calculations.
- **Mask Layer-2 Information** - This option is applied to ECMP and LAG index hash calculations.
- **Mask MPLS label information** - This option is applied to ECMP and LAG index hash calculations.
- **Diversification** - This option is applied to ECMP and LAG index hash calculations.
- **Hash Rotate** - This option is applied to ECMP hash calculations and to LAG index calculations.
- **Symmetric** - This option is applied to trunk hash calculations.

NOTE

The Brocade NetIron CES Series devices do not support the same options as the Brocade NetIron XMR Series and Brocade NetIron MLX Series devices. Refer to the Brocade NetIron CES and Brocade NetIron CER Link Aggregation chapter for additional information on hash calculations used for IP load sharing and LAGs on the Brocade NetIron CES Series devices.

Speculate UDP or TCP packet headers

With this option set, the packet headers following IPv4 headers are used for the ECMP and LAG index hash calculations even if the packet is not a TCP or UDP packet. If the packet is a non-fragmented, no-IP options, TCP or UDP packet, the TCP or UDP ports are used for hash calculations unless the **load-balance mask ip** or **load-balance mask ipv6** commands are used. This behavior is disabled by default and can be enabled using the following command.

```
device(config)# load-balance force-l4-hashing all
```

Syntax: [no] load-balance force-l4-hashing [**all** | **slot-number** | **slot-number np-id**]

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

NOTE

Problems can occur with the **Ping** and **Traceroute** functions when this option is enabled.

Masking Layer 3 and Layer 4 information

Masking in networking means that a specific header field is used for hashing. With the Layer 3 and Layer 4 masking option set, the following values can be masked during ECMP and LAG index hash calculations: TCP or UDP source and destination port information, source and destination IP address, IPv4 protocol ID, and IPv6 next header.

When used with the **load-balance force-l4-hashing** command, the **load-balance mask ip** command takes precedence. The masking option can be set using the following commands for IPv4 addresses.

```
device(config)# load-balance mask ip src-l4-port all
```

Syntax: [no] load-balance mask ip [**dst-ip** [*slot number* | **all** | **pre-symmetriclb**] | **src-ip** [*slot number* | **all** | **pre-symmetriclb**] | **dst-l4-port** |[*slot number* | **all**] | **src-l4-port** [*slot number* | **all**] | **protocol** [*slot number* | **all**]]

Use the **src-l4-port** option when you want to mask the Layer 4 source port.

Use the **dst-l4-port** option when you want to mask the Layer 4 destination port.

Use the **src-ip** option when you want to mask the source IP address. The **src-ip** keyword contains the **pre-symmetriclb** option that masks the source IP address before symmetric load balancing can occur.

Use the **dst-ip** option when you want to mask the destination IP address. The **dst-ip** keyword contains the **pre-symmetriclb** option that masks the destination IP address before symmetric load balancing can occur.

Use the **protocol** option when you want to mask the IPv4 protocol ID.

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

The masking option can be set using the following commands for IPv6 addresses.

```
device(config)# load-balance mask ipv6 src-l4-port all
```

Syntax: [no] load-balance mask ipv6 [**dst-ip** [*slot number* | **all** | **pre-symmetriclb**] | **src-ip** [*slot number* | **all** | **pre-symmetriclb**] | **dst-l4-port** |[*slot number* | **all**] | **src-l4-port** [*slot number* | **all**] | **next-hdr** [*slot number* | **all**]]

Except for the **next-hdr** option, the command options described for the **load-balance mask ip** command are valid for the **load-balance mask ipv6**.

Use the **next-hdr** option when you want to mask the IPv6 next header.

Use the **src-ip** option when you want to mask the source IPv6 address. The **src-ip** keyword contains the **pre-symmetriclb** option that masks the source IPv6 address before symmetric load balancing can occur. The symmetric load balancing can be either static or dynamic LAG load balancing.

Use the **dst-ip** option when you want to mask the destination IPv6 address. The **dst-ip** keyword contains the **pre-symmetriclb** option that masks the destination IPv6 address before symmetric load balancing can occur.

The **[no] load-balance mask ip** and **[no] load-balance mask ipv6** commands are disabled by default.

NOTE

The *Masking Layer 3 and Layer 4 information* feature supports both static and dynamic LAG load balancing.

Masking Layer 2 information

With the **load-balance mask ethernet** command set, the following Layer 2 values can be masked during ECMP and LAG index hash calculations: source and destination MAC address, VLAN, Ethertype, and Inner VLAN. To mask Layer 2 information, use the **load-balance mask ethernet** command, as shown in the following.

```
device(config)# load-balance mask ethernet sa-mac all
```

Syntax: [no] load-balance mask ethernet [**sa-mac** | **da-mac** | **vlan** | **etype** | **inner-vlan**] [**all** | **slot-number** | **slot-number np-id**]

Use the **sa-mac** option when you want to mask the source MAC address.

Use the **da-mac** option when you want to mask the destination MAC address.

Use the **vlan** option when you want to mask the VLAN ID.

Use the **etype** option when you want to mask the Ethertype

Use the **inner-vlan** option when you want to mask the inner VLAN ID.

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

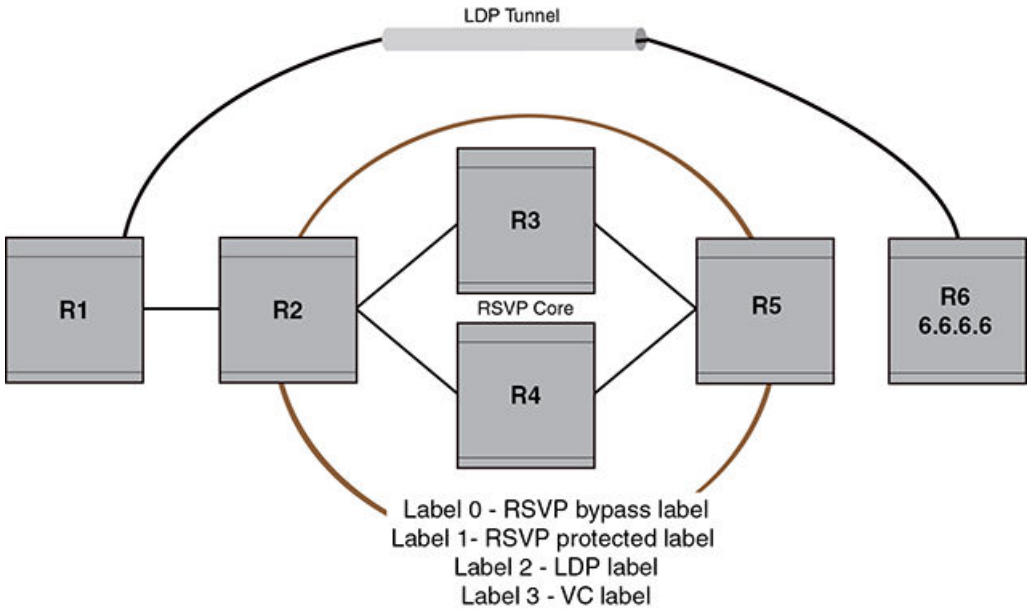
Configuring mask option for load balancing

In an MPLS network, when the L2VPN is configured using a LDP tunnel, which in turn is using a RSVP bypass tunnel, then the packets will include four labels. The four labels are:

- RSVP bypass label - Label 0 which is the outermost MPLS label
- RSVP protected label - Label 1
- LDP label - Label 2
- VC label - Label 3 which is the innermost MPLS label

In the Figure 198, all the packets routed between the routers, R2 and R5 include four MPLS labels which are masked for calculating the ECMP and LAG index hash value.

FIGURE 195 L2VPN packets over a LDP tunnel



To mask the MPLS labels, enter the following command.

```
device(config)# load-balance mask mpls label0 all
```

Syntax: [no] load-balance mask mpls [**label0** | **label1** | **label2** | **label3**] [**all** | *slot-number* | *slot-number np-id*]

Use the **label0** option to mask MPLS Label 0, which is the innermost MPLS label in a packet.

Use the **label1** option to mask MPLS Label 1, which is the next innermost MPLS label in a packet from MPLS Label 2.

Use the **label2** option to mask MPLS Label 2, which is the next innermost MPLS label in a packet with four labels or the outermost MPLS label in a packet with three labels.

Use the**label3** option to mask MPLS Label 3, which is the outermost MPLS label in a packet with four labels.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

Displaying MPLS masking information

To display the masking information, enter the following command.

```
device# show load-balance mask mpls
Mask MPLS options -
  Mask MPLS Label0 is enabled on -
  No Slots
  Mask MPLS Label1 is enabled on -
  No Slots
  Mask MPLS Label2 is enabled on -
  No Slots
```

```
Mask MPLS Label3 is enabled on -
All Slots
```

Table 88 describes the output parameters of the **show load-balance mask mpls** command.

TABLE 88 Output parameters of the show load-balance mask mplscommand

Field	Description
Slot	Shows the slot of the interface on which the MPLS masking is enabled.
Mask MPLS Label	Shows whether or not the following labels are masked. • Label0 - Shows if the Label 0 is masked on the interface. • Label1 - Shows if the Label 1 is masked on the interface. • Label2 - Shows if the Label 2 is masked on the interface. • Label3 - Shows if the Label 3 is masked on the interface.

To display current running configuration, enter the following command.

```
device# show running-config
!
load-balance mask mpls label3 all
```

!

Hash diversification for LAGs and IP load balancing

In a multi-stage network a traffic flow will normally use the same LAG port or same path (for IP load balancing) at each stage. The Hash Diversification feature works within an earlier stage of the hash calculation than the hash rotate feature. Using the**load-balance hash-diversify** command, you can provide a unique hash diversify value to a device, or a sub-set of ports on a device. This unique value is used in calculation of the ECMP and LAG index hash. Consequently, instead of a traffic flow always following the same port group or path, it will be distributed over different LAG or ECMP members. To apply hash diversification, use the following command.

```
device(config)# load-balance hash-diversify random all
```

Syntax: [no] load-balance hash-diversify [**number** | **random** | **slot**] [**all** | **slot-number** | **slot-number np-id**]

You can set the unique hash diversify value using one of the following options:

The *number* option allows you to specify a value from 0 - 255.

The **random** option directs the CPU to generate a random number for each packet processor and program it as the hash diversification value.

The **slot** option specifies the slot ID as the has diversification number.

The default value for the diversification number is 0 and the **no** version of the command resets the value to 0 regardless of any value previously set. Also, the most recent command added overrides any previous instances of the command. For example, if the **random** option is entered first and is then followed by the **slot** option, the value of the slot ID for the specified slot will be used.

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

This option can also be used in a multi-stage network to avoid the same traffic flow to always use one path of an ECMP or the same LAG member index at each stage. Using the hash rotate function the same set of traffic flows forwarded out of one LAG member or ECMP path to the next router can be distributed across different paths of the LAG member or ECMP path to the next router.

Hash rotate for LAGs and IP load balancing

The hash rotate function provides another option (in addition to hash diversification) for diversifying traffic flow in a multi-stage network. Using this feature, the ECMP hash index can be rotated by a specified number of bits after it has been calculated. This allows path selection within IP load balancing to be more diverse.

To configure hash rotate to LAG index calculations, enter a command such as the following.

```
device(config)# load-balance hash-rotate 3 all
```

Syntax: [no] load-balance hash-rotate rotate-number [**all** | **slot-number** | **slot-number np-id**]

The *rotate-number* value specifies number of bits between 0 and 7 that you want to rotate the ECMP hash index value.

The **all** option applies the command to all ports within the device.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

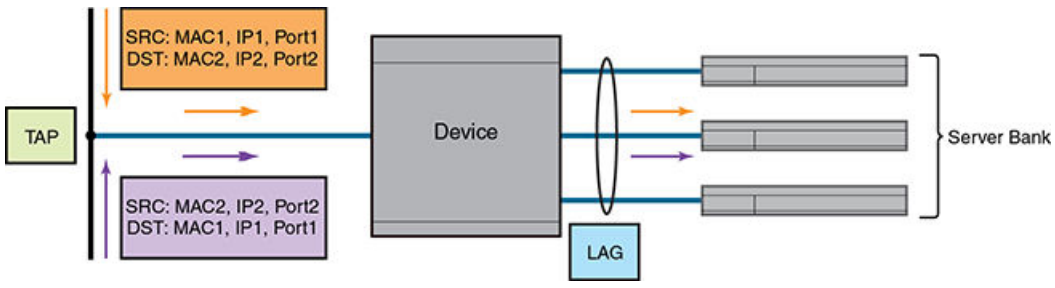
NOTE

The hash diversification and hash rotate features can be applied separately or together. Depending on your network configuration, either or both of these features may need to be configured.

Symmetric load balancing for LAGs

For many monitoring and security applications, bidirectional conversations flowing through the system must be carried on the same port of a LAG. For Network Telemetry applications, network traffic is tapped and sent to the Brocade devices, which can load balance selected traffic to the application servers downstream. Each server analyzes the bidirectional conversations. Therefore, the Brocade devices must enable symmetric load balancing to accomplish bidirectional conversations. In addition, firewalls between the Brocade devices can be configured to allow the bidirectional conversations per link of the LAG. These applications also require symmetric load balancing on the LAGs between the Brocade devices. Figure 199 depicts the symmetric load balancing for LAGs feature.

FIGURE 196 Symmetric load balancing for LAGs



NOTE

The symmetric load balancing option is applicable only for Brocade NetIron MLX Series and Brocade NetIron XMR Series devices. The Brocade NetIron CER Series and Brocade NetIron CES Series devices load balance all traffic on the LAGs symmetrically. Therefore, the Brocade NetIron CER Series and Brocade NetIron CES Series devices do not support the symmetric load balancing commands.

With the symmetric load balancing option set, the trunk hash calculation is determined using all or a combination of the following parameters: MAC source and destination addresses, IPv4 source and destination addresses, IPv6 source and destination addresses, TCP or UDP source and destination port information, inner MAC source and destination addresses, inner IPv4 source and destination addresses, and inner IPv6 source and destination addresses.

To enable the symmetric load balancing option on an interface, enter commands such as the following.

```
device(config)# load-balance symmetric ethernet 2
device(config)# load-balance symmetric ip all
device(config)# load-balance symmetric ipv6 2
device(config)# load-balance symmetric l4_ip 2
device(config)# load-balance symmetric l4_ipv6 2
device(config)# load-balance symmetric inner_ethernet 2
device(config)# load-balance symmetric inner_ip 2
device(config)# load-balance symmetric inner_ipv6 2
```

Syntax: [no] load-balance symmetric ethernet | **ip** | **ipv6** | **l4_ip** | **l4_ipv6** | **inner_ethernet** | **inner_ip** | **inner_ipv6** | **packet** [**all** | *slot-number* | *slot-number np-id*]

The **ethernet** option specifies the Ethernet header fields.

The **ip** option specifies the IP header fields.

The **ipv6** option specifies the IPv6 header fields.

The **l4_ip** option specifies the Layer 4 IP fields

The **l4_ipv6** option specifies the Layer 4 IPv6 fields.

The**inner_ethernet** option specifies the inner Ethernet fields.

The **inner_ip** option specifies the inner IP fields.

The **inner_ipv6** option specifies the inner IPv6 fields.

The **packet** option specifies all the packet fields.

The **all** option applies the command to all ports within the router.

Specifying a slot number using the *slot-number* variable limits the command to an individual module.

Specifying a slot number and a network processor ID using the *slot-number* and *np-id* variables limits the command to the ports supported by the specified network processor on the specified interface module.

The **no** option is used to turn off the previously enabled symmetric load balancing option.

Displaying symmetric load balancing information

To display the symmetric load balancing information for the interface, enter the following command.

```
device# show load-balance symmetric-options
Symmetric Ethernet options -
  Symmetric Ethernet is enabled on -
  Slot 2
  Slot 3
Symmetric IP options -
  Symmetric IP is enabled on -
  All Slots
Symmetric IPv6 options -
  Symmetric IPV6 is enabled on -
  Slot 1
  Slot 2
Symmetric IP Layer 4 IP options -
  Symmetric Layer 4 IP is enabled on -
  Slot 2
Symmetric IPv6 Layer 4 IPV6 options -
```

```
    Symmetric Layer 4 IPV6 is enabled on -
    Slot 2
  Symmetric INNER Ethernet options -
    Symmetric INNER Ethernet is enabled on -
    Slot 2
  Symmetric INNER IP options -
    Symmetric INNER IP is enabled on -
    Slot 2
  Symmetric INNER IPV6 options -
    Symmetric INNER IPV6 is enabled on -
    Slot 2
```

Syntax: show load-balance symmetric-options ethernet | ip | ipv6 | l4_ip | l4_ipv6 | inner_ethernet | inner_ip | inner_ipv6 | packet

Table 89 describes the output parameters of the **show load-balance symmetric-options** command.

TABLE 89 Output parameters of the show load-balance symmetric-options command

Field	Description
Slot	Shows the slot number of the interface on which the symmetric load balancing option is enabled.
Symmetric options	Shows whether or not the symmetric option is enabled on the following interfaces: • Symmetric Ethernet options - Shows if the symmetric option is enabled on the Ethernet interface. • Symmetric IP options - Shows if the symmetric option is enabled on the IP interface. • Symmetric IPv6 options - Shows if the symmetric option is enabled on the IPv6 interface. • Symmetric Layer 4 IP options - Shows if the symmetric option is enabled on the Layer 4 IP interface. • Symmetric Layer 4 IPv6 options - Shows if the symmetric option is enabled on the Layer 4 IPv6 interface. • Symmetric INNER Ethernet options - Shows if the symmetric option is enabled on the inner Ethernet interface. • Symmetric INNER IP options - Shows if the symmetric option is enabled on the inner IP interface. • Symmetric INNER IPv6 options - Shows if the symmetric option is enabled on the inner IPv6 interface. • Symmetric packet options - Shows if the symmetric option is enabled on all the interfaces.

How IP load sharing works

On the Brocade device, IP load sharing is done by the hardware. If there is more than one path to a given destination, a hash is calculated based on the source MAC address, destination MAC address, source IP address, destination IP address, VLAN-ID (if applicable), IPv4 protocol number, IPv6 next header and TCP/UDP source port and destination port if the packet is also a TCP/UDP packet. This hash is used to select one of the paths.

Changing the maximum number of load sharing paths

By default, IP load sharing allows IP traffic to be balanced across up to four equal path. You can change the maximum number of paths that the Brocade device supports to a value between 2 and 32.

NOTE

The maximum number of paths supported by the BR-MLX-10Gx24-DM module is 16.

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 Brocade device has six next-hop routers, set the maximum paths value to six.

NOTE
If the setting for the maximum number of paths is lower than the actual number of equal-cost paths, the software does not use all the paths for load sharing.

To change the maximum number of load sharing paths, enter the following command:

```
device(config)# ip load-sharing 32
```

Syntax: [no] ip load-sharing number

The *number* parameter specifies the number of ECMP load sharing paths. Enter a value between 2 and 32 for *number* to set the maximum number of paths. The default value is 4.

NOTE
A new **maximum-paths use-load-sharing** command was introduced under the BGP configuration that allows support for BGP routes in IP load sharing but does not enable BGP multipath load sharing.

Response to path state changes

If one of the load-balanced paths becomes unavailable, the IP route table in hardware is modified to stop using the unavailable path. The traffic is load balanced between the available paths using the same hashing mechanism described above. (Refer to How IP load sharing works on page 773.)

Configuring IRDP

The Brocade device uses ICMP Router Discovery Protocol (IRDP) to advertise the IP addresses of its device interfaces to directly attached hosts. IRDP is disabled by default. You can enable it globally or on individual ports.

Consider the following when you enable or disable IRDP globally:

- If you enable IRDP globally, all ports use the default values for the IRDP parameters.
- If you leave IRDP disabled globally but enable it on individual ports, you also can configure the IRDP parameters on an individual port basis.

NOTE
You can configure IRDP parameters only an individual port basis. To do so, IRDP must be disabled globally and enabled only on individual ports. You cannot configure IRDP parameters if the feature is globally enabled.

When IRDP is enabled, the Brocade device periodically sends Router Advertisement messages out the IP interfaces on which the feature is enabled. The messages advertise the Brocade device's IP addresses to directly attached hosts who listen for the messages. In addition, hosts can be configured to query the Brocade device for the information by sending Router Solicitation messages.

Some types of hosts use the Router Solicitation messages to discover their default gateway. When IRDP is enabled, the Brocade device responds to the Router Solicitation messages. Some clients interpret this response to mean that the Brocade device is the default gateway. If another router is actually the default gateway for these clients, leave IRDP disabled on the Brocade device.

IRDP uses the following parameters. If you enable IRDP on individual ports rather than globally, you can configure these parameters on an individual port basis. The IRDP parameters are as follows:

- **Packet type** - The Brocade device can send Router Advertisement messages as IP broadcasts or as IP multicasts addressed to IP multicast group 224.0.0.1. The packet type is IP broadcast.

- **Maximum message interval and minimum message interval** - When IRDP is enabled, the Brocade device sends the Router Advertisement messages every 450 - 600 seconds by default. The time within this interval that the Brocade device selects is random for each message and is not affected by traffic loads or other network factors. The random interval minimizes the probability that a host will receive Router Advertisement messages from other routers at the same time. The interval on each IRDP-enabled Brocade device interface is independent of the interval on other IRDP-enabled interfaces. The default maximum message interval is 600 seconds. The default minimum message interval is 450 seconds.
- **Hold time** - Each Router Advertisement message contains a hold time value. This value specifies the maximum amount of time the host should consider an advertisement to be valid until a newer advertisement arrives. When a new advertisement arrives, the hold time is reset. The hold time is always longer than the maximum advertisement interval. Therefore, if the hold time for an advertisement expires, the host can reasonably conclude that the router interface that sent the advertisement is no longer available. The default hold time is three times the maximum message interval.
- **Preference** - If a host receives multiple Router Advertisement messages from different routers, the host selects the router that sent the message with the highest preference as the default gateway. The preference can be a number from 4294967296 to 4294967295. The default is 0.

Enabling IRDP globally

To globally enable IRDP, enter the following command.

```
device(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.

```
device(config)# interface ethernet 1/3
device(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 and multicast** parameter specifies the packet type the Brocade device uses to send Router Advertisement.

- **broadcast** - The Brocade device sends Router Advertisement as IP broadcasts. This is the default.
- **multicast** - The Brocade device sends Router Advertisement as multicast packets addressed to IP multicast group 224.0.0.1.

The **holdtime***seconds* parameter specifies how long a host that receives a Router Advertisement from the Brocade device should consider the advertisement to be valid. When a host receives a new Router Advertisement message from the Brocade device, the host resets the hold time for the Brocade device to the hold time specified in the new advertisement. If the hold time of an advertisement expires, the host discards the advertisement, concluding that the router interface that sent the advertisement is no longer available. The value must be greater than the value of the **maxadvertinterval** parameter and cannot be greater than 9000. The default is three times the value of the **maxadvertinterval** parameter.

The **maxadvertinterval** parameter specifies the maximum amount of time the Brocade device waits between sending Router Advertisements. You can specify a value from 1 to the current value of the **holdtime** parameter. The default is 600 seconds.

The **minadvertinterval** parameter specifies the minimum amount of time the Brocade device can wait between sending Router Advertisements. The default is three-fourths (0.75) the value of the **maxadvertinterval** parameter. If you change the **maxadvertinterval** parameter, the software automatically adjusts the **minadvertinterval** parameter to be three-fourths the new value of the **maxadvertinterval** parameter. If you want to override the automatically configured value, you can specify an interval from 1 to the current value of the **maxadvertinterval** parameter.

The **preference***number* parameter specifies the IRDP preference level of the Brocade device. If a host receives Router Advertisements from multiple routers, the host selects the router interface that sent the message with the highest interval as the host's default gateway. The valid range is 4294967296 to 4294967295. The default is 0.

Configuring UDP broadcast and IP helper parameters

Some applications rely on client requests sent as limited IP broadcasts addressed to the UDP's application port. If a server for the application receives such a broadcast, the server can reply to the client. Routers do not forward subnet directed broadcasts, so the client and server must be on the same network for the broadcast to reach the server. If the client and server are on different networks (on opposite sides of a router), the client's request cannot reach the server.

To configure the Brocade device to forward client 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 Brocade device forwards client requests for any of the application ports the Brocade device is enabled to forward to the helper address.

Forwarding support for the following application ports is enabled by default:

- bootps (port 67)
- dns (port 53)
- tftp (port 69)
- time (port 37)
- netbios-ns (port 137)
- netbios-dgm (port 138)
- tacacs (port 65)

NOTE
The application names are the names for these applications that the Brocade device recognizes, and might not match the names for these applications on some third-party devices. The numbers listed in parentheses are the UDP port numbers for the applications. The numbers come from RFC 1340.

NOTE
As shown above, forwarding support for BootP or DHCP is enabled by default. If you are configuring the Brocade device to forward BootP or DHCP requests, refer to Configuring BootP or DHCP forwarding parameters on page 778.

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 Brocade device is not also disabled.

Enabling forwarding for a UDP application

If you want the Brocade device to forward client requests for UDP applications that the Brocade device does not forward by default, you can enable forwarding support for the port. To enable forwarding support for a UDP application, use either of the following methods. You also can disable forwarding for an application using these methods.

> NOTE
> You also must configure a helper address on the interface that is connected to the clients for the application. The Brocade device cannot forward the requests unless you configure the helper address. Refer to Configuring an IP helper address on page 779.

To enable the forwarding of specific UDP application broadcasts, enter the following command.

```
device(config)# ip forward-protocol udp bootpc
```

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)
- 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.

```
device(config)# no ip forward-protocol udp well known application port number
```

This command disables forwarding of specific UDP application requests to the helper addresses configured on Brocade device interfaces.

Configuring an IP helper address

To forward a client's broadcast request for a UDP application when the client and server are on different networks, you must configure a helper address on the interface connected to the client. Specify the server's IP address or the subnet directed broadcast address of the IP subnet the server is in as the helper address.

You can configure up to 16 helper addresses on each interface. You can configure a helper address on an Ethernet port or a virtual interface.

To configure a helper address on interface 2 on chassis module 1, enter the following commands.

```
device(config)# interface e 1/2
device(config-if-e1000-1/2)# ip helper-address 10.95.7.6
```

The commands in this example change the CLI to the configuration level for port 1/2, then add a helper address for server 10.95.7.6 to the port. If the port receives a client request for any of the applications that the Brocade device is enabled to forward, the Brocade device forwards the client's request to the server.

Syntax: [no] ip helper-address ip-addr

The *ip-addr* command specifies the server's IP address or the subnet directed broadcast address of the IP subnet the server is in.

Configuring BootP or DHCP forwarding parameters

A host on an IP network can use BootP or DHCP to obtain its IP address from a BootP or DHCP server. To obtain the address, the client sends a BootP or DHCP request. The request is a subnet directed broadcast and is addressed to UDP port 67. A limited IP broadcast is addressed to IP address 255.255.255.255 and is not forwarded by the Brocade device or other IP routers.

When the BootP or DHCP client and server are on the same network, the server receives the broadcast request and replies to the client. However, when the client and server are on different networks, the server does not receive the client's request, because the Brocade device does not forward the request.

You can configure the Brocade device to forward BootP or DHCP requests. To do so, configure a helper address on the interface that receives the client requests, and specify the BootP or DHCP server's IP address as the address you are helping the BootP or DHCP requests to reach. Instead of the server's IP address, you can specify the subnet directed broadcast address of the IP subnet the server is in.

> NOTE
> The IP subnet configured on the port which is directly connected to the device sending a BootP or DHCP request, does not have to match the subnet of the IP address given by the DHCP server.

BootP or DHCP forwarding parameters

The following parameters control the Brocade device's forwarding of BootP or DHCP requests:

- **Helper address** - The BootP or DHCP server's IP address. You must configure the helper address on the interface that receives the BootP or DHCP requests from the client. The Brocade device cannot forward a request to the server unless you configure a helper address for the server.
- **Gateway address** - The Brocade device places the IP address of the interface that received the BootP or DHCP request in the request packet's Gateway Address field (sometimes called the Router ID field). When the server responds to the request, the server sends the response as a unicast packet to the IP address in the Gateway Address field. (If the client and server are directly attached, the Gateway ID field is empty and the server replies to the client using a unicast or broadcast packet, depending on the server.) By default, the Brocade device uses the lowest-numbered IP address on the interface that receives the request as the Gateway address. You can override the default by specifying the IP address you want the Brocade device to use.
- **Hop Count** - Each router that forwards a BootP or DHCP packet increments the hop count by 1. Routers also discard a forwarded BootP or DHCP request instead of forwarding the request if the hop count is greater than the maximum number of BootP or DHCP hops allowed by the router. By default, the Brocade device forwards a BootP or DHCP request if its hop count is four or less, but discards the request if the hop count is greater than four. You can change the maximum number of hops the Brocade device will allow to a value from 1 - 15.

  > NOTE
  > The BootP or DHCP hop count is not the TTL parameter.

Configuring an IP helper address

The procedure for configuring a helper address for BootP or DHCP requests is the same as the procedure for configuring a helper address for other types of UDP broadcasts. Refer to Configuring an IP helper address on page 777.

Changing the IP address used for stamping BootP or DHCP requests

When the Brocade device forwards a BootP or DHCP request, the Brocade device "stamps" the Gateway Address field. The default value the Brocade device uses to stamp the packet is the lowest-numbered IP address configured on the interface that received the request.

The BootP or DHCP stamp address is an interface parameter. Change the parameter on the interface that is connected to the BootP or DHCP client.

To change the IP address used for stamping BootP or DHCP requests received on interface 1/1, enter commands such as the following.

```
device(config)# int e 1/1
device(config-if-e1000-1/1)# ip bootp-gateway 10.157.22.26
```

These commands change the CLI to the configuration level for port 1/1, then change the BootP or DHCP stamp address for requests received on port 1/1 to 10.157.22.26. The Brocade device will place this IP address in the Gateway Address field of BootP or DHCP requests that the Brocade device receives on port 1/1 and forwards to the BootP or DHCP server.

Syntax: [no] ip bootp-gateway ip-addr

If the **ip bootp-source-address** command is configured on the interface where the BootP or DHCP request is received, then the configured address will be used as the source IP address for the forwarded packets.

```
device(config-if-e1000-1/1)# ip bootp-source-address 10.157.22.26
```

Syntax: [no] ip bootp-source-address ip-addr

Changing the maximum number of hops to a BootP relay server

Each BootP or DHCP request includes a field Hop Count field. The Hop Count field indicates how many routers the request has passed through. When the Brocade device receives a BootP or DHCP request, the Brocade device looks at the value in the Hop Count field:

- If the hop count value is equal to or less than the maximum hop count the Brocade device allows, the Brocade device increments the hop count by one and forwards the request.
- If the hop count is greater than the maximum hop count the Brocade device allows, the Brocade device discards the request.

  NOTE
  The BootP or DHCP hop count is not the TTL parameter.

To modify the maximum number of BootP or DHCP hops, enter the following command.

```
device(config)# bootp-relay-max-hops 10
```

This command allows the Brocade device to forward BootP or DHCP requests that have passed through up to ten previous hops before reaching the Brocade device.

Syntax: [no] bootp-relay-max-hops 1-15

Default: 4

Filtering Martian addresses

Martian addresses are obviously invalid host or network addresses. They commonly are sent by improperly configured systems on the network. Martian address filtering allows the system to automatically filter out those invalid addresses. When Martian address filtering is enabled, the BGP protocol applies the Martian address filters to all in-bound routes as received from all neighbors. Unlike BGP protocol, IGP protocols will rely on the RTM (routing table manager) to do the route filtering.

If no match is found, the route is accepted. This will be the case for almost all routes. If a match is found, the route is discarded (default action - deny), unless the action is set to permit. Martian address filtering is in addition to normal BGP in-bound route policies.

To enable Martian address filtering, enter the following command.

```
device(config)#  ip martian filtering-on
```

Syntax: [no] ip martian [**vrf name**] **filtering-on**

The **vrf** *name* option applies martian filtering to a specified VRF.

> **NOTE**
> Martian address filtering is disabled by default.

When Martian address filtering is first enabled, the device will automatically load the following default Martian addresses:

* 0.0.0.0/8
* 10.0.0.0/8
* 127.0.0.0/8
* 172.16.0.0/12
* 192.168.0.0/16
* 224.0.0.0/4
* 240.0.0.0/4

Adding, deleting or modifying Martian addresses

As described previously, there are a set number of Martian addresses that are loaded by default when Martian addressing is enabled. You can add, subtract or modify addresses that are filtered by martian addressing. Although there is no limit of the number of martian address can be configured, it's expected the size of martian address list should be small, generally less than 100. If the user adds a new martian address after routes are already learnt, they will be taken out of the routing table. Likewise if the user removes a martian address after routes are deleted from the routing table, they should be put back into the routing table.

To add an address to the Martian filtering list, use a command such as the following.

```
device(config)# ip martian 192.168.0.0/16
```

Syntax: [no] ip martian [**vrf name**] **destination-prefix/prefix-length** [**permit**]

The *destination-prefix/prefix-length* variable specifies the address and the prefix range to apply the martian filtering to. The matching rule is for prefix range match. It includes exact match, or with a longer prefix length match. For example, if the Martian address rule is 192.168.0.0/16, then routes 192.168.0.0/16, and 192.168.1.0/24 are matches. However route 192.0.0.0/8 is not a match.

The **vrf** *name* option applies the modification to the martian filtering list to a specified VRF.

The **no command** removes an address from the martian filtering list.

The **[permit]** option changes the default action of a martian address filter to permit. In this case, a route matches the "permit" martian address is accepted by the routing table manager. This option is only used if a user wants to allow a prefix "hole" in an otherwise denied martian address.

The default Martian addresses are described in: Filtering Martian addresses on page 780

Examples

To remove a user defined Martian address or a system default Martian address, use the "no" form of the command.

```
device(config)# no ip martian 0.0.0.0/8
```

The following example configuration, creates a "hole" for 192.168.1.0/24 in the martian address 192.168.0.0/16.

```
device(config)# ip martian 192.168.1.0/24 permit
device(config)# ip martian 192.168.0.0/16
```

To display the currently configured Martian addresses refer to Displaying martian addressing information on page 806.

IPv6 Over IPv4 tunnels in hardware

To enable communication between the isolated IPv6 domains using the IPv4 infrastructure, you can configure IPv6 over IPv4 tunnels.

> NOTE
> The Brocade NetIron CES Series and Brocade NetIron CER Series currently do not support IPv6 over IPv4 tunneling.

Brocade devices support the following IPv6 over IPv4 tunneling in hardware mechanisms:

- Manually configured tunnels
- Automatic 6to4 tunnels

In general, a manually configured tunnel establishes a permanent link between routers in IPv6 domains, while the automatic tunnels establish a transient link that is created and taken down on an as-needed basis. (Although the feature name and description may imply otherwise, some configuration is necessary to set up an automatic tunnel.) Also, a manually configured tunnel has explicitly configured IPv4 addresses for the tunnel source and destination, while the automatic tunnels have an explicitly configured IPv4 address for the tunnel source and an automatically generated address for the tunnel destination.

These tunneling mechanisms require that the router at each end of the tunnel run both IPv4 and IPv6 protocol stacks. The routers running both protocol stacks, or dual-stack routers, can interoperate directly with both IPv4 and IPv6 end systems and routers.

The following features are not supported for IPv6 tunnel configuration at this time:

- Keep-alive
- Hitless upgrade
- Tunnels over MPLS or GRE

Configuring a IPv6 IP tunnel

To configure a IPv6 IP Tunnel, configure the following parameters:

- CAM Restrictions
- Maximum Number of Tunnels (optional)
- Tunnel Interface
- Source Address or Source Interface for the Tunnel
- Destination address for the Tunnel

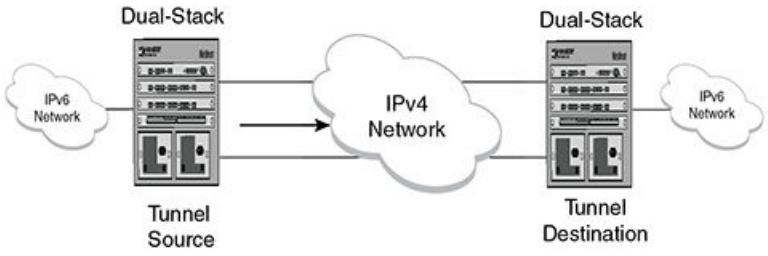
- IPv6 Encapsulation
- IP address for the Tunnel
- TTL Value (optional)
- TOS Value (optional)
- MTU Value (optional)

NOTE
Do not forward packets from one type of tunnel to another type of tunnel in XPP. Packets may not be routed properly.

Configuring a manual IPv6 tunnel

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnel mechanism if you need a permanent and stable connection.

FIGURE 197 Manually configured tunnel



Configuration notes on manual tunnels:

- The tunnel mode should be **ipv6ip** indicating that this is an IPv6 manual tunnel.
- Both source and destination addresses needs to be configured on the tunnel.
- On the remote side you need to have exactly opposite source/destination pair.
- The tunnel destination should be reachable through the IPv4 backbone.
- The ipv6 address on the tunnel needs to be configured for the tunnel to come up.
- The tunnel source can be an IP address or interface name.
- Manual tunnels provide static point-point connectivity.
- Static routing on top of the tunnel is supported.
- IPv6 routing protocols including OSPFv3 and RIPing on top of the tunnel are supported.

NOTE
IPv6 IS-IS is not supported on top of the tunnel.

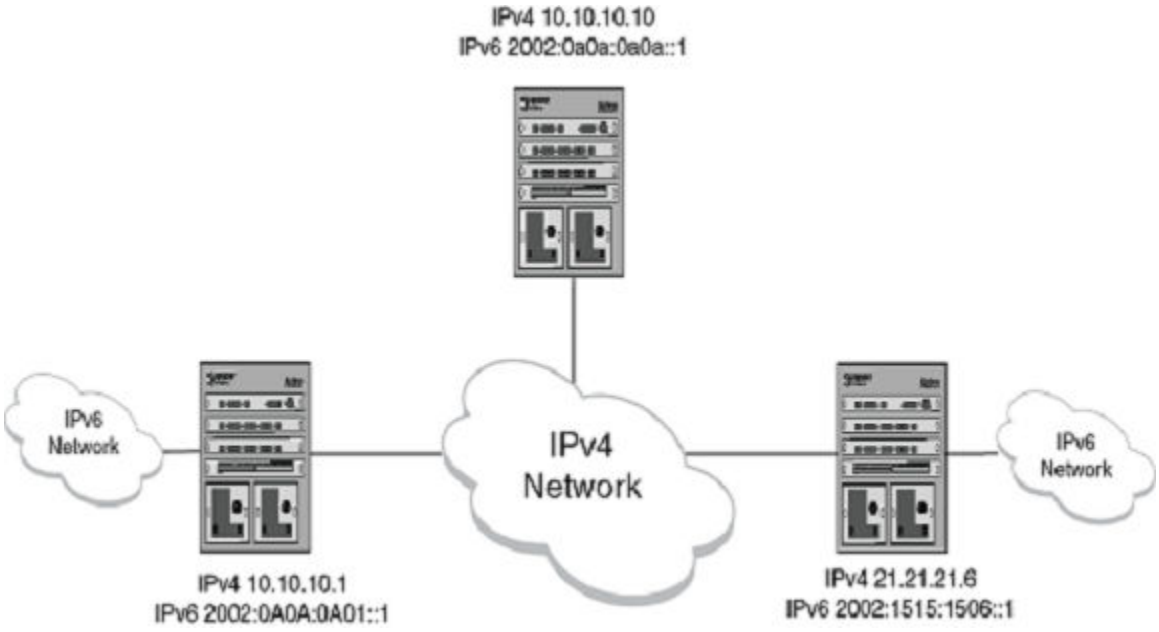
Configuring an automatic 6to4 tunnel

An automatic 6to4 tunnel establishes a transient link between IPv6 domains, which are connected by an IPv4 backbone. When needed, a device on which an automatic 6to4 tunnel is configured in one domain can establish a tunnel with another similarly configured device in another domain. When no longer needed, the devices take down the tunnel.

Instead of a manually configured tunnel destination, an automatic 6to4 tunnel constructs a globally unique 6to4 prefix, which determines the tunnel destination. The 6to4 prefix has the following format:

2002:*ipv4-address* ::/48

When two domains need to communicate, a device creates a tunnel using the 6to4 prefix. The software automatically generates the 6to4 prefix by concatenating a configured static IPv6 prefix of 2002 with the destination device's globally unique IPv4 address. (Each device in an IPv6 domain that needs to communicate over an automatic 6to4 tunnel must have one globally unique IPv4 address, from which the globally unique 6to4 prefix is constructed.) After the communication ends, the tunnel is taken down.



Configuration notes on 6to4tunnels:

- This tunnel treats the IPv4 infrastructure as a virtual non-broadcast link and support multipoint connectivity.
- Tunnel mode must be configured as **ipv6ip 6to4.**
- Tunnel source must be configured.
- Tunnel destination is not configured on 6to4 tunnel explicitly, as the destination is specified as part of static nexthop or BGP nexthop.
- Static route with **2002::/16** MUST be configured.
- IPv6 address with **2002:A.B.C.D::/48** must be configured for the tunnel to come up (A.B.C.D is the tunnel source IP address).
- You can have 6to4 tunnel with multiple nexthops depending on the IPv6 nexthop used to forward the packets.
- With 6to4 tunnels, you can only use routing protocols (that is BGP+) that specify the nexthop in the configuration.
- OSPFv3, IPv6 IS-IS and RIPng are not supported on the 6to4 tunnels.
- Static routes can be used with 6to4 tunnels. If you use a static route to configure the nexthop, you MUST enable nexthop recursion in the system (ipv6 route next-hop-recursion).
- The 6to4 tunnel tries to resolve all the nexthops and programs the cam and pram entries needed. The IPv4 address in the nexthop should be reachable through the IPv4 network.

In the below configuration:

- – 10.10.10.1 is the tunnel source IP address
  – 10.10.10.10 is the static nexthop
  – 21.21.21.6 is I-BGP nexthop
  – 22.22.22.6 is E-BGP nexthop

Static route Nexthop example:

- Create a static route pointing to the tunnel.

```
device(config) #ipv6 route 2002::/16 tunnel 2 // Mandatory for 6to4 Configuration
device(config) #ipv6 route next-hop-recursion // Mandatory with static nexthop
device(config)# ipv6 route 3001::/64 2002:0a0a:0a0a::1 // Static Nexthop: 10.10.10.10
```

- Create a Source Interface - The remote node needs to have a similar route pointing to this node.

```
device(config)# interface ethernet 1/1
device(config-if-e10000-1 /1)ip address 10.10.10.1 255.255.255.0
```

- Create a 6to4 Tunnel configuration.

```
device(config) interface tunnel 2
device(config-tnif-2) tunnel mode ipv6ip 6to4
device(config-tnif-1) tunnel source 10.10.10.1
device(config-tnif-1) ipv6 address 2002:0a0a:0a01::1/64
```

: I-BGP Nexthop.

```
device(config) router bgp
device(config-bgp) local-as 100
device(config-bgp) neighbor 2002:1515:1506::1 remote-as 100 // BGP Nexthop: 21.21.21.6
device(config-bgp)# address-family ipv4 unicast
device(config-bgp)# no neighbor 2001:DB8:1506::1 activate
device(config-bgp)# exit-address-family
device(config-bgp)# address-family ipv4 multicast
device(config-bgp)# exit-address-family
device(config-bgp)# address-family ipv6 unicast
device(config-bgp)# neighbor 2002:1515:1506::1 activate
device(config-bgp)# exit-address-family
```

: E-BGP Nexthop.

```
device(config)# router bgp
device(config-bgp)# local-as 100
device(config-bgp)# neighbor 2002:1616:1606::1 remote-as 101 // BGP Nexthop: 22.22.22.6
device(config-bgp)# neighbor 2002:1616:1606::1 ebgp-multihop
device(config-bgp)# address-family ipv4 unicast
device(config-bgp)# no neighbor 2002:1515:1506::1 activate
device(config-bgp)# exit-address-family
device(config-bgp)# address-family ipv4 multicast
device(config-bgp)# exit-address-family
device(config-bgp)# address-family ipv6 unicast
device(config-bgp)# neighbor 2002:1616:1606::1 activate
device(config-bgp)# exit-address-family
```

Configuring the maximum number of tunnels supported

You can configure the device to support a specified number of tunnels using the following command.

```
device(config)# system-max ip-tunnels 512
device(config)# write memory
```

Syntax: [no] system-max ip-tunnels number

The *number* variable specifies the number of IPv6 tunnels that can be supported on the Brocade device. The permissible range is 1 - 512. The default value is 256.

NOTE
You must write this command to memory and perform a system reload for this command to take effect.

Configuring a tunnel interface

To configure a tunnel interface, use a the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)
```

Syntax: [no] interface tunnel tunnel id

The *tunnel-id* variable is numerical value that identifies the tunnel being configured.Possible range is from 1 to the maximum configured tunnels in the system.

Configuring a source address or source interface for a tunnel interface

To configure a source address for a specific tunnel interface, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel source 10.0.8.108
```

To configure a source interface for a specific tunnel interface, enter the following command.

```
device(config)# interface tunnel 100
device(config-tnif-100)tunnel source ethernet 3/1
```

Syntax: [no] tunnel source ip-address | port-no

You can specify either of the following:

The *ip-address* variable is the source IP address being configured for the specified tunnel.The *port-no* variable is the source slot/port of the interface being configured for the specified tunnel. When you configure a source interface, there must be at least one IP address configured on that interface. Otherwise, the interface will not be added to the tunnel configuration and an error message like the following will be displayed: "Error - Tunnel source interface 3/1 has no configured ip address.

It can be a physical or virtual interface (ve).

Configuring a destination address for a tunnel interface

To configure a destination address for a specific tunnel interface, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel destination 10.108.5.2
```

Syntax: [no] tunnel destination ip-address

The *ip-address* variable is destination IP address being configured for the specified tunnel.

Configuring a tunnel interface for IPv6 encapsulation

To configure a specified tunnel interface for IPv6 encapsulation, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel mode ipv6ip
```

Syntax: [no] tunnel mode ipv6ip 6to4 | auto-tunnel

The **6to4** parameter specifies automatic tunneling using 6 to 4.

The **auto-tunnel** parameter specifies automatic tunnel using IPv4 compatible ipv6 address.

Configuring an IP address for a tunnel interface

To configure an IP address for a specified tunnel interface, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)ipv6 address 2001:0a0a:0a01::1/64
```

Syntax: [no] ipv6 address ipv6-address

The *ipv6-address* variable is the IPv6 address being configured for the specified tunnel interface.

Configuring a TTL value

This is an optional parameter that allows you to set the Time-to-Live value for the outer IP header of the IPv6 tunnel packets.

To configure the TTL value, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel ttl 100
```

Syntax: [no] tunnel ttl ttl-value

The *ttl-value* variable specifies a TTL value for the outer IP header. Possible values are 1 - 255. The default value is 255.

Configuring a TOS value

This is an optional parameter that allows you to set the TOS value for the outer IP header of the GRE tunnel packets.

To configure the TOS value, enter the following command.

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel tos 100
```

Syntax: [no] tunnel ttl tos-value

The *tos-value* variable specifies a TOS value for the outer IP header. Possible values are 1 - 255. The default value is 0.

Configuring IPv6 session enforce check

You can enable the IPv6 session enforce check by using the **ipv6-session-enforce-check** command. When an IPv6 packet arrives and this feature is enabled, the system tries to match the IPv6 packet source and destination address pair with the tunnel configured destination and source pair. If the pairs do not match, the packet is dropped in hardware.

To configure the IPv6 session enforce check, go to the IP tunnel policy context and enter the **ipv6-session-enforce-check** command.

```
device(config)#ip-tunnel-policy
device(config-ip-tunnel-policy)#ipv6-session-enforce-check
```

Syntax: [no] ipv6-session-enforce-check

To disable the IPv6 session enforce check, use the **no** form of this command. This command is disabled by default. You might have to write the configuration to memory and reload the system when the configuration of this command is changed because a one-time creation of a source-ingress CAM partition is necessary. The system prompts you if the memory write and reload are required.

The first-time execution of certain commands necessitates the creation of a source-ingress CAM partition, after which you write to memory and reload. These commands are **gre-session-enforce-check** , **ipv6-session-enforce-check** , and **accounting-enable** . After this CAM partition is created, it is not necessary to follow either of the other two commands with a memory write and reload.

NOTE

The ipv6-sessions-enforce-check is not supported for 6to4 automatic tunnels.

Configuring a maximum MTU value for a tunnel interface

This command allows you to set an MTU value for packets entering the tunnel. Packets that exceed either the default MTU value of 1480 bytes or the value that you set using this command are sent back to the source.

The following command allows you to change the MTU value for packets transiting "tunnel 1".

```
device(config)# interface tunnel 1
device(config-tnif-1)tunnel mtu 1500
```

Syntax: [no] tunnel mtu packet-size

The *packet-size* variable specifies the maximum MTU size in bytes for the packets transiting the tunnel.

> NOTE
> To prevent packet loss after the 20 byte IP header is added, make sure that any physical interface that is carrying IPv6 tunnel traffic has an IP MTU setting at least 24 bytes greater than the tunnel MTU setting.

Displaying IPv6 tunneling information

You can display IPv6 Tunneling Information using the **show ip-tunnels** , **show ipv6 interface** , **show ipv6 route** and **show interface tunnel** commands as shown in the following:

Displaying tunnel information

For example, to tunnel information for tunnel 2, enter the following command at any level of the CLI.

```
device# show ip-tunnels 2
IPv6 tnnl 2 UP   : src_ip 10.211.2.1, dst_ip 10.212.2.1
   TTL 255, TOS 0, NHT 0, MTU 1480
```

Syntax: show ip tunnels number

The *number* parameter indicates the tunnel interface number for which you want to display information.

This display shows the following information.

TABLE 90 Show IP tunnel display information

This field...	Displays...
IPv6 tnnl *UP* / *DOWN*	The status of the tunnel interface can be one of the following: • up - The tunnel interface is functioning properly. • down - The tunnel interface is not functioning and is down.
src_ip	The tunnel source can an IPv4 address.
dst_ip	The tunnel destination can an IPv4 address.
TTL	The TTL value configured for the outer IP header. Possible values are 1 - 255.
TOS	The TOS value configured for the outer IP header. Possible values are 1 - 255.
NHT	The nextHop Table index value.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying tunnel interface information

For example, to display status and configuration information for tunnel interface 1, enter the following command at any level of the CLI.

```
device# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
```

```
Hardware is Tunnel
Tunnel source ethernet 3/5
Tunnel destination is not configured
Tunnel mode ipv6ip auto-tunnel
No port name
MTU 1500 bytes
```

Syntax: show interfaces tunnel number

The *number* parameter indicates the tunnel interface number for which you want to display information.

This display shows the following information.

TABLE 91 IPv6 tunnel interface information

This field...	Displays...
Tunnel interface status	The status of the tunnel interface can be one of the following: • up - The tunnel interface is functioning properly. • down - The tunnel interface is not functioning and is down.
Line protocol status	The status of the line protocol can be one of the following: • up - The line protocol is functioning properly. • down - The line protocol is not functioning and is down.
Hardware is tunnel	The interface is a tunnel interface.
Tunnel source	The tunnel source can be one of the following: • An IPv4 address • The IPv4 address associated with an interface or port.
Tunnel destination	The tunnel destination can an IPv4 address.
Tunnel mode	The tunnel mode can be one the following: • ipv6ip auto-tunnel - Indicates an automatic IPv4-compatible tunnel. • ipv6ip 6to4 - Indicates an automatic 6to4 tunnel.
Port name	The port name configured for the tunnel interface.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying interface level IPv6 settings

To display Interface level IPv6 settings for tunnel interface 1, enter the following command at any level of the CLI.

```
device# show ipv6 inter tunnel 1
Interface Tunnel 1 is up, line protocol is up
  IPv6 is enabled, link-local address is fe80::3:4:2 [Preferred]
  Global unicast address(es):
    1001::1 [Preferred],  subnet is 1001::/64
    1011::1 [Preferred],  subnet is 1011::/64
  Joined group address(es):
    ff02::1:ff04:2
    ff02::5
    ff02::1:ff00:1
    ff02::2
    ff02::1
  MTU is 1480 bytes
  ICMP redirects are enabled
  No Inbound Access List Set
  No Outbound Access List Set
  OSPF enabled
```

The display command above reflects the following configuration.

```
device# show running-config interface tunnel 1
!
```

```
interface tunnel 1
 port-name ManualTunnel1
 tunnel mode ipv6ip
 tunnel source loopback 1
 tunnel destination 10.1.1.1
 ipv6 address fe80::3:4:2 link-local
 ipv6 address 1011::1/64
 ipv6 address 1001::1/64
 ipv6 ospf area 0
```

Displaying IP information

You can display the following IP configuration information statistics:

- **Global IP parameter settings** - refer to Displaying global IP configuration information on page 789.
- **IP interfaces** - refer to Displaying IP interface information on page 790.
- **ARP entries** - refer to Displaying ARP entries on page 793.
- **Static ARP entries** - refer to Displaying ARP entries on page 793.
- **IP forwarding cache** - refer to Displaying the forwarding cache on page 795.
- **IP route table** - refer to Displaying the IP route table on page 796.
- **IP traffic statistics** - refer to Displaying IP traffic statistics on page 800.

The sections below describe how to display this information.

In addition to the information described below, you can display the following IP information:

- **RIP information**
- **OSPF information**
- **BGP4 information**
- **PIM information**

Displaying global IP configuration information

To display IP configuration information, enter the following command at any CLI level.

```
device> show ip
Global Settings
  IP CAM Mode: dynamic IPVPN CAM Mode: static
  ttl: 64, arp-age: 10, bootp-relay-max-hops: 4, icmp-error-rate: 400
  IP Router-Id: 10.5.5.5
  enabled : UDP-Broadcast-Forwarding ICMP-Redirect Source-Route Load-Sharing
  RARP BGP4 OSPF
  disabled: Directed-Broadcast-Forwarding drop-arp-pending-packets IRDP Proxy
  -ARP RPF-Check RPF-Exclude-Default RIP IS-IS VRRP VRRP-Extended VSRP
Configured Static Routes: 31
Configured Static Mroutes: 30
```

Syntax: show ip

> NOTE
> This command has additional options, which are explained in other sections in this guide, including the sections below this one.

This display shows the following information.

TABLE 92 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 Brocade device. If the packet's TTL value is higher than the value specified in this field, the device drops the packet. To change the maximum TTL, refer to Changing the TTL threshold on page 740.
arp-age	The ARP aging period. This parameter specifies how many minutes an inactive ARP entry remains in the ARP cache before the device ages out the entry. To change the ARP aging period, refer to Changing the ARP aging period on page 715.
bootp-relay-max-hops	The maximum number of hops away a BootP server can be located from the device and still be used by the device's clients for network booting. To change this value, refer to Changing the maximum number of hops to a BootP relay server on page 779.
router-id	The 32-bit number that uniquely identifies the device. By default, the router ID is the numerically lowest IP interface configured on the device. To change the router ID, refer to Changing the router ID on page 709.
enabled	The IP-related protocols that are enabled on the device.
disabled	The IP-related protocols that are disabled on the device.

Displaying IP interface information

To display IP interface information, enter the following command at any CLI level.

```
Interface       IP-Address      OK?  Method Status     Protocol VRF
eth 3/10        10.25.25.3      YES  NVRAM  down       down     default-vrf
eth 3/19        10.11.11.3      YES  NVRAM  up         up       default-vrf
eth 3/20        10.33.32.1      YES  NVRAM  up         up       default-vrf
mgmt 1          10.25.106.12    YES  NVRAM  up         up       default-vrf
loopback 1      10.5.5.5        YES  NVRAM  up         up       default-vrf
```

Syntax: show ip interface [**ethernet slot/port**] | [**loopback num**] | [**ve num**]

This display shows the following information.

TABLE 93 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.

TABLE 93 CLI display of interface IP configuration information (continued)

This field...	Displays...
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI, but have not saved the configuration, the entry for the interface in the Method field is "manual".
Status	The link status of the interface. If you have disabled the interface with the **disable** command, the entry in the Status field will be "administratively down". Otherwise, the entry in the Status field will be either "up" or "down".
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field will be "up". Otherwise the entry in the protocol field will be "down".
VRF	Specifies the VRF type applied to the interface.

Displaying IP interface information for a specified interface

To display detailed IP information for a specific interface, enter a command such as the following.

```
device# show ip interface ethernet e 3/1
Interface Ethernet 3/1 (80)
  port enabled
  port state: UP
  ip address: 10.1.1.2/24
  Port belongs to VRF: default
  encapsulation: ETHERNET, mtu: 1500
  MAC Address 0004.80a0.4050
  directed-broadcast-forwarding: disabled
  No inbound ip access-list is set
  No outbound ip access-list is set
  No Helper Addresses are configured.
  RPF mode: None  RPF Log: Disabled
  0 unicast RPF drop    0 unicast RPF suppressed drop
  RxPkts:  1200  TxPkts:  1200
  RxBytes: 60000 TxBytes: 60000
```

NOTE

Interface counters (received packets and received bytes) are not supported on the Brocade NetIron CES Series or the Brocade NetIron CER Series devices. These values will always be 0.

The Brocade device software supports IPv4 and IPv6 packet and byte counters. The contents of these counters is displayed for a defined port as the result of the show ip interface ethernet command. In the above example, the fields in bold text display this content. Table 94 describes each of the fields that display interface counter statistics.

TABLE 94 Interface counter display statistics

This field...	Displays...
Interface	The interface that counter statistics are being displayed for.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Displaying interface counters for all ports

The Brocade device supports IPv4 and IPv6 packet and byte counters. The contents of these counters can be displayed for all ports on a device or per-port. Output from the **show ip interface ethernet** command has been enhanced to include packet and byte counter information on a per-port basis. This is described in Displaying interface counters for all ports.

NOTE
The **shop ip interface ethernet** command is not supported on Brocade NetIron CES Series and Brocade NetIron CER Series devices. The command is supported only on the Brocade NetIron XMR Series and the Brocade MLXe Series devices.

Commands have been added under IPv4 and IPv6 to display the interface counters for all ports on a device. The following example uses the **show ip interface counters** command to display to packet and byte counter information for all ports.

```
device# show ip interface counters
Interface      RxPkts     TxPkts      RxBytes    TxBytes
eth 3/1         1200       1200       600000     60000
eth 3/2         500        500        25000      25000
```

Syntax: show ip interface counters

Default byte counters include the 20-byte per-packet Ethernet overhead. You can configure an Brocade device to exclude the 20-byte per-packet Ethernet overhead from byte accounting by configuring the **vlan-counter exclude-overhead** command.Displaying IP interface information for a specified interface on page 791 describes each of the fields that display interface counter statistics.

TABLE 95 Interface counter display statistics

This field...	Displays...
Interface	The interface that counter statistics are being displayed for.
RxPkts	The number of packets received at the specified port.
TxPkts	The number of packets transmitted from the specified port.
RxBytes	The number of bytes received at the specified port.
TxBytes	The number of bytes transmitted from the specified port.

Clearing the interface counters

Use the following command to clear all interface counters on a device.

NOTE
The **clear ip interface counters** command is available for the Brocade NetIron CES Series and the Brocade NetIron CER Series devices; however, the counters are not supported and the values will always be 0.

```
device# clear ip interface counters
```

Syntax: clear ip interface counters

Use the following command to clear the interface counters for a specified port.

```
device# clear ip interface ethernet 3/2
```

Syntax: clear ip interface ethernet port-number

The *port-number*variable specifies the slot and port number that you want to clear the interface counters for.

Displaying interface name in Syslog

By default an interface's slot number (if applicable) and port number are displayed when you display Syslog messages. You can display the name of the interface instead of its number by entering a command such as the following.

```
device(config)# ip show-portname
```

This command is applied globally to all interfaces on the Brocade device.

Syntax: [no] Ip show-portname

When you display the messages in the Syslog, you see the interface name under the Dynamic Log Buffer section. The actual interface number is appended to the interface name. For example, if the interface name is "lab" and its port number is "2", you see "lab2" displayed as in the example below.

```
device# 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 Brocade device. The static ARP table contains the user-configured ARP entries. An entry in the static ARP table enters the ARP cache when the entry's interface comes up.

The tables require separate display commands.

Displaying the ARP cache

To display the contents of the ARP cache, enter the following command at any CLI level.

```
device# show arp
Total number of ARP entries: 5
     IP Address          MAC Address         Type        Age     Port
1     10.95.6.102        0800.5afc.ea21      Dynamic     0         6
2     10.95.6.18         00a0.24d2.04ed      Dynamic     3         6
3     10.95.6.54         00a0.24ab.cd2b      Dynamic     0         6
4     10.95.6.101        0800.207c.a7fa      Dynamic     0         6
5     10.95.6.211        00c0.2638.ac9c      Dynamic     0         6
6     10.30.30.15        none                Pending     0         v1
```

Syntax: show arp [**ethernet slot/port** | **mac-address xxxx.xxxx.xxxx** [**mask**] | **ip-addr** [**ip-mask**]] [**num**] [| **begin expression** | **exclude expression** | **include expression**]

The **ethernet***slot/portnum* parameter lets you restrict the display to entries for a specific port.

The **mac-address***xxxx.xxxx.xxxx* parameter lets you restrict the display to entries for a specific MAC address.

The *mask* parameter lets you specify a mask for the **mac-address***xxxx.xxxx.xxxx* parameter to display entries for multiple MAC addresses. Specify the MAC address mask as fs and 0s, where fs are significant bits.

The *ip-addr* and *ip-mask* parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

> NOTE
> The *ip-mask* parameter and *mask* parameter perform different operations. The *ip-mask* parameter specifies the network mask for a specific IP address, whereas the *mask* parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

The *num* parameter lets you display the table beginning with a specific entry number.

> NOTE
> The entry numbers in the ARP cache are not related to the entry numbers for static ARP table entries.

This display shows the following information. The number in the left column of the CLI display is the row number of the entry in the ARP cache. This number is not related to the number you assign to static MAC address entries in the static ARP table.

TABLE 96 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 Brocade device learned the entry from an incoming packet. • Static - The Brocade device loaded the entry from the static ARP table when the device for the entry was connected to the Brocade device. • Pending - The Brocade device added the entry to the ARP table and is in the process of sending a series of ARP requests to determine if it is a valid entry.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the table. To display the ARP aging period, refer to Displaying global IP configuration information on page 789. To change the ARP aging interval, refer to Changing the ARP aging period on page 715. **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.

```
device# show ip static-arp
Total no. of entries: 4
  Index   IP Address          MAC Address         Port      VLAN  ESI
  1       10.1.1.1             0001.0001.0001      1/1
  2       10.6.6.2             0002.0002.0002      1/2
  3       10.6.6.7             1111.1111.1111      2/1...
          Ports  : ethe 2/1 to 2/7 ethe 3/1 to 3/2
  4       10.7.7.7             0100.5e42.7f40      3/3
```

This example shows four static entries, one of which is multi-port. Multi-port static ARP entries are supported only on the Brocade NetIron XMR Series and Brocade NetIron MLX Series devices. Note that for multi-port entries the Port column shows a single port number followed by an ellipsis; the full list of ports associated with that ARP entry is displayed on the following line.

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 793.

TABLE 97 CLI display of static ARP table

This field...	Displays...
Index	The number of this entry in the table.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.

TABLE 97 CLI display of static ARP table (continued)

This field...	Displays...
Port	The port attached to the device the entry is for. In the case of a multi-port static ARP, this will display a single port followed by an ellipsis, and the full list of ports will be displayed on the line below.
VLAN	VLAN associated with this entry, if any.
ESI	Ethernet Service Instance (ESI) associated with this entry, if any.

Displaying the forwarding cache

To display the IP Forwarding Cache for directly connected hosts, enter the following command.

```
device> 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**.

```
device>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   10.1.0.0      DIRECT       0000.0000.0000    PU      2/5     n/a      0
2   10.2.0.0      DIRECT       0125.0a57.1c02    D       3/5     n/a      0
3   10.7.7.3       DIRECT       0000.0000.0000    PU      4/2     12       1
```

You also use the **rconsole** to display the IP Forwarding Cache for network entries.

```
device>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   10.1.1.0/24      DIRECT       0000.0000.0000    PC              n/a      0
3   10.40.40.0/24    10.2.1.10    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 98 CLI display of IP forwarding cache

This field...	Displays...
IP Address	The IP address of the destination.
Next Hop	The IP address of the next-hop router to the destination. This field contains either an IP address or the value DIRECT. DIRECT means the destination is either directly attached or the destination is an address on this device. For example, the next hop for loopback addresses and broadcast addresses is shown as DIRECT.
MAC	The MAC address of the destination.

TABLE 98 CLI display of IP forwarding cache (continued)

This field...	Displays...
	NOTE If the entry is type U (indicating that the destination is this device), the address consists of zeroes.
Type	The type of host entry, which can be one or more of the following: • D - Dynamic • P - Permanent • F - Forward • U - Us • C - Complex Filter • W - Wait ARP • I - ICMP Deny • K - Drop • R - Fragment • S - Snap Encap
Port	The port through which this device reaches the destination. For destinations that are located on this device, the port number is shown as "n/a".
VLAN	Indicates the VLANs the listed port is in.
Pri	The QoS priority of the port or VLAN.

Dual Active Console

The Dual Active Console command enables the standby terminal console to mirror the features of the active console, such that the standby console appears as active console itself. Hence, you can manage the system from either active or standby console and it will not be necessary to switch the console cable after the active-standby management module switchover.

To enable this feature, enter the following command,

```
device(config)#dual-active-console
device(config)#wr mem
Write startup-config done.
device(config)#
```

To disable this feature, enter the following command,

```
device(config)#no dual-active-console
device(config)#wr mem
Write startup-config done.
device(config)#
```

Displaying the IP route table

To display the IP route table, enter the **show ip route** command at any CLI level.

```
device# show ip route
Total number of IP routes: 4
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
        Destination        Gateway         Port        Cost       Type Uptime
1       10.0.0.0/24        DIRECT          eth 1/1     0/0        D    45m18s
2       10.10.0.0/24        DIRECT          eth 1/2     0/0        D    1h0m
3       10.20.0.0/24        10.0.0.2        eth 1/1     1/1        S    13m18s
4       10.30.0.0/24       10.0.0.2        eth 1/1     1/1        S    2m42s
```

Syntax: show ip route num | [**ip-addr** [**ip-mask**] [**debug** | **detail** | **longer**]] | **connected** | **bgp** | **isis** | **ospf** | **rip** | **static** | [**summary**] | **nexthop** [**nexthop_id** [**ref-routes**]] | [**begin expression** | **exclude expression** | **include expression**]

The *num* option display the route table entry whose row number corresponds to the number you specify. For example, if you want to display the tenth row in the table, enter "10".

The *ip-addr* parameter displays the route to the specified IP address.

The *ip-mask* parameter lets you specify a network mask or, if you prefer CIDR format, the number of bits in the network mask. If you use CIDR format, enter a forward slash immediately after the IP address, then enter the number of mask bits (for example: 10.157.22.0/24 for 10.157.22.0 255.255.255.0).

The **longer, detail, and 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 Brocade device.

The **ospf** option displays the OSPF routes.

The **rip** option displays the RIP routes.

The **isis** option displays the RIP routes.

The **static** option displays only the static IP routes.

The **nexthop** option displays next-hop information for all next hops in the routing table or for a specific entry.

Showing route details by IP address

You can display detailed information about a route by providing the IP address and using the **detail** option, as the following example illustrates.

```
device>show ip route 10.1.1.2 detail
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       10.1.1.0/24        DIRECT          eth 1/15    0/0           D    7h11m
        Nexthop Entry ID:14, Paths: 1, Ref_Count:1/1
1       10.1.1.0/24        10.1.1.2        eth 1/15    115/20        IL2  7h11m
        10.1.1.0/24        10.0.0.18      eth 4/11    115/20        IL2  7h11m
        10.1.1.0/24        10.0.0.30      eth 4/7     115/20        IL2  7h11m
        10.1.1.0/24        10.0.0.34      eth 4/14    115/20        IL2  7h11m
        Nexthop Entry ID:68343, Paths: 4, Ref_Count:8/21
D:Dynamic  P:Permanent  F:Forward  U:Us  C:Connected Network
W:Wait ARP  I:ICMP Deny  K:Drop  R:Fragment  S:Snap Encap N:CamInvalid
Module S1:
      IP Address         Next Hop        MAC              Type  Port  Vlan  Pri
      10.1.1.0/24          DIRECT           0000.0000.0000     PC    n/a           0
      OutgoingIf  ArpIndex PPCR_ID   CamLevel   Parent  DontAge Index
       eth 1/15     65535    1:2        1              0    69203192   38
      U_flags   Entry_flags  Age   Cam:Index          Trunk_fid      Ecmp_count
       0000e220                0     0x1a8fc (L3, right)   0x00000(     0) 0
        CAM Entry Flag: 00000003H
        PPCR : 1:2 CIDX: 0x1a8fc (L3, right) (IP_NETWORK: 0x68703)
         PPCR : 1:1 CIDX: 0x1a8fc (L3, right) (IP_NETWORK: 0x68703)
```

Syntax: show ip route ip_addr detail

The IP address can be just the IP address but can also include shorthand for the mask: ip-address/prefix-length.

Using the summary option

The **summary** option displays a summary of the information in the IP route table. After the **summary** keyword, the pipe symbol (|) points to three options for modifying the presentation of the summary information, as follows:

- **begin** lets you start the display with the first matching line.
- **exclude** lets you exclude matching lines from the display.
- **include** lets you include matching lines in the display.

The default routes are displayed first.

Using the connected option

Here is an example of how to use the **connected** option. To display only the IP routes that go to devices directly attached to the Brocade device.

```
device(config)# show ip route connected
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination       Gateway           Port   Cost   Type   Uptime
1     10.157.22.0/24     0.0.0.0           4/11    1       D      1h0m
```

Notice that the route displayed in this example has "D" in the Type field, indicating the route is for a directly connected device.

Using the static option

Here is an example of how to use the **static** option. To display only the static IP routes.

```
device(config)# show ip route static
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
      Destination       Gateway           Port   Cost   Type  Uptime
1     10.144.33.11/32  10.157.22.12     1/1    2      S     1h0m
```

Notice that the route displayed in this example has "S" in the Type field, indicating the route is static.

Using the longer option

Here is an example of how to use the **longer** option. To display only the routes for a specified IP address and mask, enter a command such as the following.

```
device(config)# show ip route
10.159.0.0/16 longer
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
   Destination       Gateway          Port   Cost   Type  Uptime
52 10.159.38.0/24   10.95.6.101      1/1    1      S     45m18s
53 10.159.39.0/24   10.95.6.101      1/1    1      S     1h0m
54 10.159.40.0/24   10.95.6.101      1/1    1      S     45m18s
55 10.159.41.0/24   10.95.6.101      1/1    1      S     1h0m
56 10.159.42.0/24   10.95.6.101      1/1    1      S     13m18s
```

This example shows all the routes for networks beginning with 209.159. The mask value and **longer** parameter specify the range of network addresses to be displayed. In this example, all routes within the range 209.159.0.0 - 209.159.255.255 are listed.

Using the summary option

The **summary** option displays a summary of the information in the IP route table. The following is an example of the output from this command.

```
device# show ip route summary
IP Routing Table - 35 entries:
  6 connected, 28 static, 0 RIP, 1 OSPF, 0 BGP, 0 ISIS, 0 MPLS
```

```
Number of prefixes:
/0: 1 /16: 27 /22: 1 /24: 5 /32: 1
```

Syntax: show ip route summary

In this example, the IP route table contains 35 entries. Of these entries, 6 are directly connected devices, 28 are static routes, and 1 route was calculated through OSPF. One of the routes has a zero-bit mask (this is the default route), 27 have a 22-bit mask, 5 have a 24-bit mask, and 1 has a 32-bit mask.

Using the nexthop option

You can display next-hop information for all next hops in the routing table or for a specific entry. For the first example, use the **show ip route nexthop** command to display all the next-hop entries, and then use the option to display the next hop for a specific table entry.

```
device#show ip route nexthop
Total number of IP nexthop entries: 30; Forwarding Use: 24
       NextHopIp        Port             RefCount    ID      Age
1      0.0.0.0         mgmt 1          0/1         1536    80682
2      0.0.0.0         eth 1/15        1/1         14      80632
3      0.0.0.0         eth 1/16        1/1         15      16626
4      0.0.0.0         eth 1/18        1/1         17      16626
5      0.0.0.0         eth 1/43        1/1         42      35923
6      0.0.0.0         eth 1/47        1/1         46      80641
7      0.0.0.0         eth 2/2         1/1         49      16630
8      0.0.0.0         eth 2/4         1/1         51      16630
9      10.1.1.2        eth 1/15        0/2         68347   16620
       10.1.2.2        eth 1/18
       10.0.0.18      eth 4/11
       10.0.0.25      eth 4/9
10     10.1.1.2        eth 1/15        0/3         68352   16615
       10.0.0.6       eth 4/4
       10.0.0.10      eth 2/2
       10.0.0.21      eth 4/1
11     0.0.0.0         eth 4/1         1/1         144     16624
12     0.0.0.0         eth 4/3         1/1         146     16641
13     0.0.0.0         eth 4/4         1/1         147     16624
14     0.0.0.0         eth 4/6         1/1         149     16624
15     0.0.0.0         eth 4/7         1/1         150     16641
```

Syntax: show ip route nexthop [**nexthop_id**]

The *nexthop_id* is under the column labeled ID in the output of the **show ip route nexthop** command. For example, use nexthop ID 1536 from the first row of the preceding example to show only that entry.

```
device#show ip route nexthop 1536
       NextHopIp        Port             RefCount      ID      Age
1      0.0.0.0          mgmt 1           0/1           1536    80685
```

Displaying IP routes with nexthop ID

By using the **nexthop** option with the **ref-routes** keyword, you can display IP routes in the forwarding table that refer to the specified nexthop entry, as the following example illustrates (using nexthop ID 65575).

```
device#show ip route nexthop 65537 ref-routes
Type Codes - B:BGP D:Connected I:ISIS O:OSPF R:RIP S:Static; Cost - Dist/Metric
ISIS Codes - L1:Level-1 L2:Level-2
OSPF Codes - i:Inter Area 1:External Type 1 2:External Type 2 s:Sham Link
        Destination        Gateway         Port        Cost          Type Uptime
1       10.1.1.1/32        10.2.1.1        eth 1/11    115/10        IL2  7h51m
2       10.1.1.0/24        10.2.1.1        eth 1/11    115/10        IL2  7h51m
3       10.1.1.1/32        10.2.1.1        eth 1/11    115/40        IL2  7h51m
```

Syntax: show ip route nexthop [**nexthop_id** [**ref-routes**]]

Description of command output fields

The following table lists the information in the **show ip route** output when you use no optional arguments.

TABLE 99 CLI display of IP route table

This field...	Displays...
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The port through which this device 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 Brocade device. • R - The route was learned from RIP. • S - The route is a static route. • * - The route is a candidate default route. • O - The route is an OSPF route. Unless you use the ospf option to display the route table, "O" is used for all OSPF routes. If you do use the ospf option, the following type codes are used: • O - OSPF intra area route (within the same area). • IA - The route is an OSPF inter area route (a route that passes from one area into another). • E1 - The route is an OSPF external type 1 route. • E2 - The route is an OSPF external type 2 route.
Uptime	The amount of time since the route was last modified. The format of this display parameter may change depending upon the age of the route to include the seconds (s), minutes (m), hours (h), and days (d), as described in the following: 400d - Only days (d) displayed 20d23h - days (d) and hours (h) displayed 14h33m - hours (h) and minutes (m) displayed 10m59s - minutes (m) and seconds (s) displayed

Clearing IP routes

If needed, you can clear the entire route table or specific individual routes.

To clear all routes from the IP route table.

```
device# clear ip route
```

To clear route 10.157.22.0/24 from the IP routing table.

```
device# clear ip route 10.157.22.0/24
```

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 Brocade device, only those packets that are forwarded or generated by the CPU are included in the IP traffic statistics. Hardware forwarded packets are not included.

```
device# show ip traffic
IP Statistics
  1265602 total received, 690204 mp received, 225395 sent, 0 forwarded
  0 filtered, 0 fragmented, 0 bad header
  0 failed reassembly, 0 reassembled, 0 reassembly required
  2951 no route, 0 unknown proto, 0 no buffer, 0 other errors
ARP Statistics
  489279 total recv, 488154 req recv, 1125 rep recv, 1159 req sent, 3960 rep sent
  0 pending drop, 0 invalid source, 0 invalid dest
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 address mask, 0 address mask reply
  0 irdp advertisement, 0 irdp solicitation
Sent:
  2146 total, 0 errors, 2146 unreachable, 0 time exceed (0 mpls-response)
  0 parameter, 0 source quench, 0 redirect, 0 echo, 0 echo reply
  0 timestamp, 0 timestamp reply, 0 address mask, 0 address mask reply
  0 irdp advertisement, 0 irdp solicitation
UDP Statistics
  184784 received, 75473 sent, 110196 no port, 0 input errors
TCP Statistics
  86199 in segments, 84392 out segments, 909 retransmission, 0 input errors
ip packet list pool
pool: 237598e3, unit_size: 9362, initial_number:32, upper_limit:128
  total_number:32, allocated_number:0, alloc_failure 0
  flag: 0, pool_index:1, avail_data:27100000
ip reassembly list pool
pool: 23759783, unit_size: 23, initial_number:16, upper_limit:64
  total_number:16, allocated_number:0, alloc_failure 0
  flag: 0, pool_index:1, avail_data:270df800
ip fragments list pool
pool: 23759833, unit_size: 20, initial_number:32, upper_limit:128
  total_number:32, allocated_number:0, alloc_failure 0
  flag: 0, pool_index:1, avail_data:270e0800
```

Syntax: show ip traffic

The **show ip traffic** command displays the following information.

TABLE 100 CLI display of IP traffic statistics

This field...	Displays...
IP statistics	
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
forwarded	The total number of IP packets received by the device and forwarded to other devices.
filtered	The total number of IP packets filtered by the device.
fragmented	The total number of IP packets fragmented by this device to accommodate the IP MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.

TABLE 100 CLI display of IP traffic statistics (continued)

This field...	Displays...
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Brocade customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.
ICMP statistics The ICMP statistics are derived from RFC 792, "Internet Control Message Protocol", RFC 950, "Internet Standard Subnetting Procedure", and RFC 1256, "ICMP Router Discovery Messages". Statistics are organized into Sent and Received. The field descriptions below apply to each.	
total	The total number of ICMP messages sent or received by the device.
errors	This information is used by Brocade customer support.
unreachable	The number of Destination Unreachable messages sent or received by the device.
time exceed	The number of Time Exceeded messages sent or received by the device.
parameter	The number of Parameter Problem messages sent or received by the device.
source quench	The number of Source Quench messages sent or received by the device.
redirect	The number of Redirect messages sent or received by the device.
echo	The number of Echo messages sent or received by the device.
echo reply	The number of Echo Reply messages sent or received by the device.
timestamp	The number of Timestamp messages sent or received by the device.
timestamp reply	The number of Timestamp Reply messages sent or received by the device.
addr mask	The number of Address Mask Request messages sent or received by the device.
addr mask reply	The number of Address Mask Replies messages sent or received by the device.
irdp advertisement	The number of ICMP Router Discovery Protocol (IRDP) Advertisement messages sent or received by the device.
irdp solicitation	The number of IRDP Solicitation messages sent or received by the device.
UDP statistics	
received	The number of UDP packets received by the device.
sent	The number of UDP packets sent by the device.
no port	The number of UDP packets dropped because the packet did not contain a valid UDP port number.
input errors	This information is used by Brocade customer support.
TCP statistics The TCP statistics are derived from RFC 793, "Transmission Control Protocol".	
active opens	The number of TCP connections opened by this device by sending a TCP SYN to another device.
passive opens	The number of TCP connections opened by this device in response to connection requests (TCP SYNs) received from other devices.
failed attempts	This information is used by Brocade customer support.
active resets	The number of TCP connections this device reset by sending a TCP RESET message to the device at the other end of the connection.

TABLE 100 CLI display of IP traffic statistics (continued)

This field...	Displays...
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 Brocade customer support.
in segments	The number of TCP segments received by the device.
out segments	The number of TCP segments sent by the device.
retransmission	The number of segments that this device retransmitted because the retransmission timer for the segment had expired before the device at the other end of the connection had acknowledged receipt of the segment.

Displaying GRE tunnel information

To display information about all GRE tunnels configured on a device, enter the following command.

```
device# show gre
Total Valid GRE Tunnels : 1, GRE Session Check Enforce: FALSE
GRE tnnl 1 UP : src_ip 10.25.25.4, dst_ip 10.15.15.3
   TTL 255, TOS 0, NHT 0, MTU 1476
```

Syntax: show gre tunnel-number

The *tunnel-number* option allows you to limit the display to information about a specified tunnel.

Displaying GRE and manual IPv6 tunnel statistics

This section contains examples of the following **show** commands for GRE tunnel and manual IPv6 tunnel statistics:

- **show ip-tunnels**
- **show ip-tunnel** *tunnel ID*
- **show statistics brief tunnel** *tunnel ID*
- **show statistics tunnel** *tunnel ID*
- **show interface tunnel** *tunnel ID*

To see a list of the configured tunnels with some details of each tunnel, use the **show ip-tunnels** command as the following example illustrates. This example shows that one tunnel exists; it has IP tunnel statistics collection enabled; and neither GRE nor IPv6 session enforce are enabled.

```
device#show ip-tunnels
# of Valid Tunnels: 1, GRE Session Enforce: FALSE, IPv6 Session Enforce: FALSE
   IP Tunnel Statistics collection Enabled
GRE tnnl 1 UP src_ip 10.1.1.4, dst_ip 10.1.1.1
   TTL 255, TOS 0, NHT 0, MTU 1476
```

Syntax: show ip tunnel number

Syntax: show ip tunnels

The output of this command contains the following type of information:

TABLE 101 Show IP tunnel display information

This field...	Displays...
IPv6 tnnl x *UP* / *DOWN*	The status of the interface for manual IPv6 tunnel interface x can be one of the following: • UP - The tunnel interface is functioning properly.

TABLE 101 Show IP tunnel display information (continued)

This field...	Displays...
	• DOWN - The tunnel interface is not functioning and is down.
GRE tnnl x UP or DOWN	The status of the interface for GRE tunnel interface x can be one of the following: • UP - The tunnel interface is functioning properly. • DOWN - The tunnel interface is not functioning and is down.
GRE Session Enforce	Shows whether the global GRE session enforce feature is enabled. The output is one of the following: TRUE - the feature is enabled. FALSE - the feature is disabled.
IPv6 Session Enforce	Shows whether the global IPv6 session enforce feature is enabled. The output is one of the following: TRUE - the feature is enabled. FALSE - the feature is disabled.
IP Tunnel Statistics collection	Shows whether the collection of tunnel statistics is enabled. The enable or disable is a global setting that applies to both directions of GRE and manual IPv6 tunnels (unicast and multicast).
src_ip	The tunnel source can an IPv4 address.
dst_ip	The tunnel destination can an IPv4 address.
TTL	The TTL value configured for the outer IP header. The range for TTLs is 1 - 255.
TOS	The TOS value configured for the outer IP header. The range for TOS values is 1 - 255.
NHT	The nextHop Table index value.
MTU	The setting of the IPv6 maximum transmission unit (MTU).

Displaying brief tunnel statistics

Use the **show statistics brief tunnel** command to display the aggregate statistics for a specific tunnel or for all tunnels (in page mode). The feature combines both unicast and multicast statistics into one counter.

To display all of the brief statistics, run the **show statistics brief tunnel** command.

```
device#show statistics brief tunnel
                                    Packets
Tunnel Id  Tunnel Type  [Rcv-from-tnnl    Xmit-to-tnnl]
1          GRE          586046            287497
2          GRE          0                 0
3         IPV6-Manual 0                0
```

Syntax: show statistics brief tunnel tunnel-id

To show the brief statistics for a particular tunnel, include the optional tunnel ID. The types of information in the output are as follows:

TABLE 102 Show IP tunnel display information

This field...	Displays...
Tunnel ID	For each tunnel displayed, the Tunnel ID indicates the tunnel for which the statistics are displayed.
Tunnel Type	The tunnel type is either GRE or manual IPv6.
Packets Rcv-from-tnnl	The number of packets that have arrived from the tunnel.

TABLE 102 Show IP tunnel display information (continued)

This field...	Displays...
Packets Xmit-to-tnnl	The number of packets that have been sent to the tunnel.

Displaying tunnel statistics

Use the **show statistics tunnel** command to display the aggregate statistics (including unicast and multicast statistics) for a port-range for each packet processor (PPCR) for every LP module in the system. If you enter the command with no tunnel ID, the output displays statistics for all tunnels in page-mode. This command displays only the port ranges that have either unicast or multicast traffic and displays nothing if the LP does not have any non-zero counters. For this example, the LP2 is a four-port 10G card.

```
device#show statistics tunnel 1
                                              Packets
Tunnel Id  Tunnel Type      In-Port(s)     [Rcv-from-tnnl    Xmit-to-tnnl]
1          GRE          e2/1 - e2/2     586046           287497
                       e2/3 - e2/4    100340          150034
```

Syntax: show statistics tunnel tunnel-ID

The optional *tunnel-ID* parameter lets you specify a particular tunnel, otherwise all tunnel statistics are shown. The command output contains the following types of information.

TABLE 103 Show IP tunnel display information

This field...	Displays...
Tunnel ID	For each tunnel displayed, the Tunnel ID indicates the tunnel for which the statistics are displayed.
Tunnel Type	The tunnel type is either GRE or manual IPv6.
In-ports	The Ethernet ports traversed by the tunnel.
Packets Rcv-from-tnnl	The number of packets that have arrived from the tunnel.
Packets Xmit-to-tnnl	The number of packets that have been sent to the tunnel.

Displaying interface statistics for a tunnel

To see the interface statistics for a particular tunnel (GRE tunnel 1 in this case), use the **show interface tunnel** command, as the following illustrates.

```
device#show interface tunnel 1
Tunnel 1 is up, line protocol is up
  Hardware is Tunnel
  Tunnel source  10.30.30.1
  Tunnel destination is 10.20.20.1
  Tunnel mode gre ip
  No port name
  Internet address is: 10.50.50.4/24
  Tunnel TOS 0, Tunnel TTL 255, Tunnel MTU 1476 bytes
  Keepalive is not Enabled
  Tunnel Packet Statistics:
                Unicast Packets       Multicast Packets
    In-Port(s)   [Rcv-from-tnnl  Xmit-to-tnnl]  [Rcv-from-tnnl    Xmit-to-tnnl]
    e5/1 - e5/20   0           16511754        0             0
    e6/1 - e6/20   0           14147748        0             20195730
    e7/1 - e7/24   21493545           0        40696309      0
    e16/1 - e16/2  0            3916998        0             0
    e16/3 - e16/4  0            13476342       0
```

Syntax: show interface tunnel tunnel ID

The *tunnel ID* is the ID of a particular tunnel. The output contains the following types of information:

TABLE 104 Show IP tunnel display information

This field...	Displays...
Tunnel status	The tunnel and the line protocol can be • UP - The tunnel or line protocol is up and functioning properly. • DOWN - The tunnel or line protocol is down.
Tunnel source	The source IP address of the hardware.
Tunnel destination	The destination IP address of the hardware.
Tunnel mode	The tunnel mode is either GRE or manual IPv6.
Port name	The port name is displayed, or if no name has been configured, this fact is stated.
Internet address	The IP address of the ingress.
TOS	The TOS value configured for the outer IP header. The range for TOS values is 1 - 255.
TTL	The TTL value configured for the outer IP header. The range for TTLs is 1 - 255.
MTU	The setting of the IPv6 maximum transmission unit (MTU).
Keepalive	Shows the number of seconds for the keepalive option if it has been configured, otherwise it states "not Enabled."
In-ports	The Ethernet port numbers.
Unicast Packets	On a per port basis, this column shows the number of unicast packets that have arrived from the tunnel and the number of packets that have been transmitted to the tunnel. This count includes packets for both GRE tunnels and packets for manually configured IPv6 tunnels.
Multicast Packets	On a per port basis, this column shows the number of multicast packets that have arrived from the tunnel and the packets that have been transmitted to the tunnel. This count includes packets for both GRE tunnels and packets for manually configured IPv6 tunnels.

Displaying martian addressing information

To display Martian Addressing information, use the following command.

```
device# show ip martian
ip martian filtering on
0.0.0.0/8 deny
10.0.0.0/8 deny
127.0.0.0/8 deny
191.255.0.0/16 deny
192.0.0.0/24 deny
223.255.255.0/24 deny
240.0.0.0/4 deny
```

Syntax: show [vrf name] ip martian

You can use the **vrf** option to display martian addresses for a specific VRF.

Multiple VLAN Registration Protocol (MVRP)

Multiple VLAN Registration Protocol

Multiple VLAN Registration Protocol (MVRP) is an MRP application that provides IEEE 802.1ak-compliant VLAN pruning and dynamic VLAN creation on switch ports. It allows dynamic configuration of a VLAN over intermediate switches joining a set of access switches declaring a particular VLAN. MVRP aware switches exchange VLAN configuration information and maintains a dynamic reachability tree connecting all devices interested in a particular VLAN. MVRP VLAN pruning, using the reach ability tree, limits the scope of unnecessary broadcast and unknown unicast to a set of interested end devices only.

MVRP allows the propagation of VLAN information from device to device. With MVRP, an access switch is manually configured with all the desired VLANs for the network, and all other switches on the network learn those VLANs dynamically.

Enabling MVRP globally

MVRP must be enabled globally to allow the device to participate in the protocol.

```
device(config)# mvrp enable
```

Syntax: [no] mvrp enable

MVRP is disabled by default.

Global Timer Configuration

Use the **mvrp timer join** command to configure the join, leave and leave-all timers at global level for MVRP.

```
device(config)# mvrp timer join 400 leave 1400 leave-all 10000
```

Syntax: [no] mvrp timer join *value* **leave** *value* **leave-all** *value*

The timer values are set in milliseconds (ms).

The default values for the timers are:

- Join timer - 200ms
- Leave timer - 1000ms (Recommended value: 5000ms)
- Leave-all timer - 10000ms

The **leave** timer must be greater than or equal to twice the join timer plus 600ms. The recommended value for the **leave-all** timer is at least three times the value of **leave** timer.

The acceptable ranges for each time is:

- Join timer - 200 to 2147483600 ms
- Leave timer - 1000 to 2147483600 ms
- Leave-all timer - 10000 to 2147483600 ms

Configuring MVRP at the interface level

Before MVRP can be configured at the interface level, it must be enabled at global level first.

Enabling MVRP on an interface

Use the **mvrp enable** command to enable MVRP at the interface level.

```
device(config-if-e1000-1/1)#mvrp enable
```

Syntax: [no] mvrp enable

By default MVRP is disabled.

MVRP port applicant mode

Configuring non-participant over a port should be used for edge ports.

```
device(config-if-e1000-1/1)#mvrp applicant-mode non-participant
```

Syntax: [no] mvrp applicant-mode [**normal-participant** | **non-participant**]

Applicant mode by default is set as normal participant.

In **non-participant** mode, MVRP will not transmit PDUs on this port; therefore, no VLAN declarations will be made.

In **normal-participant** mode, MVRP will transmit PDUs on this port for making VLAN declarations.

MVRP port point to point

Configuring no point to point over a port should be used when a port is connected to a shared media device.

```
device(config-if-e1000-1/1)#mvrp point-to-point
```

Syntax: [no] mvrp point-to-point

By default, point-to-point is disabled.

MVRP interface level timers

Use the **mvrp timer join** command to configure the join, leave and leave-all timers at interface level for MVRP.

```
device(config-if-e1000-1/1)# mvrp timer join 400 leave 1400 leave-all 10000
```

Syntax: [no] mvrp timer join *value* **leave** *value* **leave-all** *value*

The timer values are set in milliseconds (ms).

The default values for the timers are:

- Join timer - 200ms
- Leave timer - 1000ms (Recommended value: 5000ms)
- Leave-all timer - 10000ms

The **leave** timer must be greater than or equal to twice the join timer plus 600ms. The recommended value for the **leave-all** timer is at least three times the value of **leave** timer.

The acceptable ranges for each time is:

- Join timer - 200 to 2147483600 ms
- Leave timer - 1000 to 2147483600 ms

- Leave-all timer - 10000 to 2147483600 ms

MVRP interface level registration-mode configuration

Configuring the registration mode of an interface for MVRP.

```
device(config-eth-1/1)#mvrp registration-mode forbidden vlan 10
```

Syntax: [no] mvrp registration-mode forbidden [*vlan-id* | *vlan-id* **to** *vlan-id*]

By default, registration mode is normal. For a static VLAN configuration, registration mode is automatically set to Fixed.

The **mvrp registration-mode forbidden** command will accept values from 1 to 4090.

MVRP configuration examples

Single interface MVRP configuration example

```
device(config)#int e 1/1
device(config-eth-1/1)#mvrp enable
device(config-eth-1/1)#mvrp registration-mode forbidden vlan 10
device(config-eth-1/1)#mvrp timer join 400 leave 1400 leave-all 10000
```

Multiple Interface (consecutive) MVRP configuration example

```
device(config)#int e 1/1 to e 1/2
device(config-mif-1/1-1/2)#mvrp enable
device(config- mif-1/1-1/2)#mvrp registration-mode forbidden vlan 10
device(config- mif-1/1-1/2)#mvrp timer join 400 leave 1400 leave-all 10000
```

Multiple Interface (non-consecutive) MVRP Configuration

```
device(config)#int e 1/1  e 1/3 e 1/5
device(config-mif-1/1,1/3,1/5)#mvrp enable
device(config-mif-1/1,1/3,1/5)#mvrp registration-mode forbidden vlan 10
device(config-mif-1/1,1/3,1/5)#mvrp timer join 400 leave 1400 leave-all 10000
```

Error messages

Deletion of dynamically learned VLAN

Assume VLAN 10 is dynamically learned over port 1/1. Now when VLAN 10 is manually removed, the following error message is displayed.

```
Error - Dynamic VLAN 10 cannot be deleted manually.
```

Deletion of dynamically created PORT-VLAN membership

Assume VLAN 10 is dynamically learned over port 1/1. Now when port 1/1 is manually removed from the VLAN, the following error message is displayed.

```
Error - Dynamically added port 1/1 cannot be deleted from VLAN 10 manually
```

Adding a VLAN to forbidden list over MVRP enabled port when it is statically configured

For example, assume port 1/1 is statically tagged to VLAN 10. Now when VLAN 10 is added to the forbidden list on an MVRP enabled port 1/1, the following error is displayed.

```
Error - Forbidden vlan configuration not allowed as VLAN 10 is statically configured on port 1/1.
```

Enabling MVRP when per VLAN instance of STP or RSTP is running

For example, assume per VLAN STP or RSTP is running. While enabling MVRP on the system following error is displayed.

```
Error - Please remove all per vlan instances of STP and RSTP.
```

Enabling MVRP while MSTP is configured

When MVRP is enabled while MSTP is running, the following error message is displayed.

```
Error - Please remove all MSTP configurations before running MVRP.
```

Enabling MVRP when metro ring is configured

When MVRP is enabled while metro ring is configured, the following error message is displayed.

```
Error - Please remove all metro-ring configurations before running MVRP
```

Enabling MVRP when ERP is configured

When MVRP is enabled while ERP is configured, the following error message is displayed.

```
Error - Please remove all ERP configurations before running MVRP
```

Enabling MVRP when VSRP is configured

When MVRP is enabled while VSRP is configured, the following error message is displayed.

```
Error - Please remove all VSRP configurations before running MVRP
```

Enabling MVRP when topology group is configured

When MVRP is enabled while a topology group is configured, the following error message is displayed:

```
Error - Please remove all topology group configurations before running MVRP
```

Enabling MVRP when VLAN group is configured

When MVRP is enabled while VLAN group is configured, the following error message is displayed.

```
Error - Please remove all VLAN group configurations before running MVRP
```

Enabling MVRP over VPLS end point port

When MVRP is configured over a port which is also a VPLS end point, the following error message is displayed.

```
Error - MVRP cannot be enabled on port 1/1 as it is configured as a VPLS end point
```

Enabling MVRP over port with non-default port-type

When MVRP is enabled over a port with non-default port-type, the following error message is displayed.

```
Error - MVRP cannot be enabled on port 1/1 as its port-type has been changed for PB-PBB network
```

Running per VLAN instance of STP or RSTP when MVRP is enabled

For example, assume per VLAN STP or RSTP is running. Now, while enabling MVRP on the system, the following error message is displayed.

```
Error - Please disable mvrp protocol before running STP on vlan 10.
```

Enabling MSTP when MVRP is configured

When MSTP is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running MSTP on vlan 10.
```

Enabling metro ring when MVRP is configured

When metro ring is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running MRP on VLAN 10.
```

Enabling ERP when MVRP is configured.

When ERP is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running ERP on VLAN 10.
```

Enabling VSRP when MVRP is configured

When VSRP is enabled while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp protocol before running VSRP on VLAN 10.
```

Enabling topology group when MVRP is configured

When a topology group is configured while MVRP is configured, the following error message is displayed.

```
Error - Please disable mvrp before configuring topology group.
```

Enabling VLAN group when MVRP is configured

When vlan-group is configured while MVRP is configured, the following error message is displayed.

```
Error - Please disable MVRP before configuring vlan group.
```

Adding port to VPLS vlan when MVRP is configured over the same port

When an MVRP enabled port is added to a VPLS VLAN, the following error message is displayed.

```
Error - Port %p cannot be a VPLS end-point, due to mvrp configuration.
```

Changing port type to non-default when MVRP is configured over the same port

When a port type of MVRP enabled port is changed to non-default, the following error message is displayed.

```
Error: Cannot change port type to non-default when MVRP is enabled on the same port.
```

Adding port to VPLS vlan when MVRP is configured over the same port

When an MVRP enabled port is added to a VPLS VLAN, the following error message is displayed.

```
Error:  Please disable mvrp before running %s over any vpls vlan.
```

Removing single spanning tree instance when MVRP is enabled

When a single STP or RSTP is disabled while MVRP is running, the following error message is displayed.

```
Error - single stp cannot be disabled when MVRP is running.
Error - single rstp cannot be disabled when MVRP is running.
```

When MVRP enabled port is added as secondary port of a trunk

When an MVRP enabled port is added as a secondary port of a trunk, the following error message is displayed.

```
Error - mvrp is enabled on secondary port 1/1.
```

When a VLAN having dynamic ports is removed from base spanning tree

For example, assume statically created VLAN 11 is dynamically learned on port 1/1. Now when VLAN 11 is removed from the base spanning tree, the following error message is displayed.

```
Error - Cannot remove spanning tree from VLAN: 11 as it has dynamically tagged ports.
```

Syslog Messages

When a VLAN is created dynamically

```
Jan  9 03:31:42:AM: MVRP: VLAN 100 dynamically added.
```

When a VLAN is removed dynamically

```
Jan  9 03:31:42:AM: MVRP: VLAN 100 dynamically removed.
```

When a VLAN is added on a port dynamically

```
Jan  9 03:31:42:AM: MVRP: Port 1/2 dynamically added to VLAN 100.
```

When a VLAN is removed from a port dynamically

```
Jan  9 03:31:42:AM: MVRP: Port 1/2 dynamically removed from VLAN 100.
```

When a dynamic Vport changes to static

```
Jan  9 03:31:42:AM: MVRP: Port VPORT type changed to static for 1/1 and VLAN 100.
```

When a dynamic VLAN changes to static

```
Jan  9 03:31:42:AM: MVRP: Dynamic VLAN 100 changed to static.
```

When VLAN creation threshold is reached

```
Jan  9 03:31:42:AM: MVRP: System threshold reached for creation of VLANs. MVRP could not add VLAN 100 on
port 1/1.
```

Logging control

The following command is available to control printing of the MVRP syslog.

Syntax: [no] Logging enable mvrp-vlan

Clear commands

The following set of commands are available with the MVRP feature.

Use the **clear mvrp statistics** command to clear MVRP port statistics for all ports.

```
device# clear mvrp statistics
```

Syntax: clear mvrp statistics

Use the **clear mvrp statistics eth** command to clear MVRP port statistics for the specific port.

```
device# clear mvrp statistics eth 1/1
```

Syntax: clear mvrp statistics eth *slot/port*

Multiple MAC Registration Protocol (MMRP)

Overview

Multiple MAC Registration Protocol (MMRP) provides a mechanism for end-stations and bridges to dynamically register or declare group membership or individual MAC addresses to bridges attached in the same LAN. Any given declaration is propagated to all application participants, and registered in each bridge on those ports that are closest to the source or sources of the declaration within the active topology. Registration of group membership information makes bridges aware that frames destined for the group MAC address concerned should only be forwarded in the direction of the registered members of the group. Therefore, forwarding of frames destined for the address associated with that group occurs only on ports on which such membership registration has been received.

MMRP networks

MMRP on NetIron provides the ability for flood containment for multi-point services in a PBB network.

Limitations

MMRP is not supported on untagged ports.

Propagation of Group Membership

MMRP uses the Registration Entries in the Filtering Database to ensure that the frames are transmitted to those ports on which group members are attached, thereby avoiding flooding of these frames on all the ports.

Any node required to receive frames for this group has to declare the attribute. When a bridge is required to receive frames for a group it will declare the group, membership to all the ports in the VLAN based on the active topology, so that it sends this information to all connected nodes. Each bridge receiving this declaration will register it on the incoming port and will in turn send it on all other ports there by propagating the declaration to all the nodes in the LAN. Such propagation will result in the formation of the reachability tree. When a bridge has to send frames to this particular group, it will be sent to ports on which the registration entry exists.

Definition of MRP protocol elements

Use of MAP context

MMRP, as defined in the standard, operates within the set of VLAN Contexts that correspond to the VLANs that are supported by the VLAN Bridged Local Area Network. The MAP Context Identifier used to identify a VLAN Context shall be equal to the VID used to identify the corresponding VLAN.

The set of ports defined to be part of the active topology for a given VLAN Context shall be equal to the set of ports for which the following are true:

- The port is a member of the member set of that VLAN; and
- The port is one of the ports that are part of the active topology for the spanning tree that supports that VLAN.

Context identification in MMRP

The ingress rules for MMRPDUs on the port receiving such frames.

- MMRP frames with no VLAN classification (that is, untagged or priority-tagged MMRPDUs) are discarded if the port is tagged port in the vlan.
- VLAN-tagged MMRP frames are classified according to the VID carried in the tag header.
- If the port is not in the member set for the MMRP frame's VLAN classification, then the frame is discarded.
- MMRPDUs transmitted by MMRP Participants are VLAN classified according to the VLAN Context associated with that Participant. The following rules apply when the MMRPDUs are transmitted.
- MMRPDUs are transmitted through a given port only if the port is the member of the VLAN.
- MMRPDUs are transmitted as VLAN-tagged frames or as untagged frames in accordance with the port is tagged or untagged Port for the VLAN concerned. Where VLAN-tagged frames are transmitted, the VID field of the tag header carries the VLAN Context Identifier value.

MMRPDU

The MMRPDU frames will use the destination MAC of 00-00-00-00-00-20 and Ether-type of 88-F6 and protocol version of 0x00

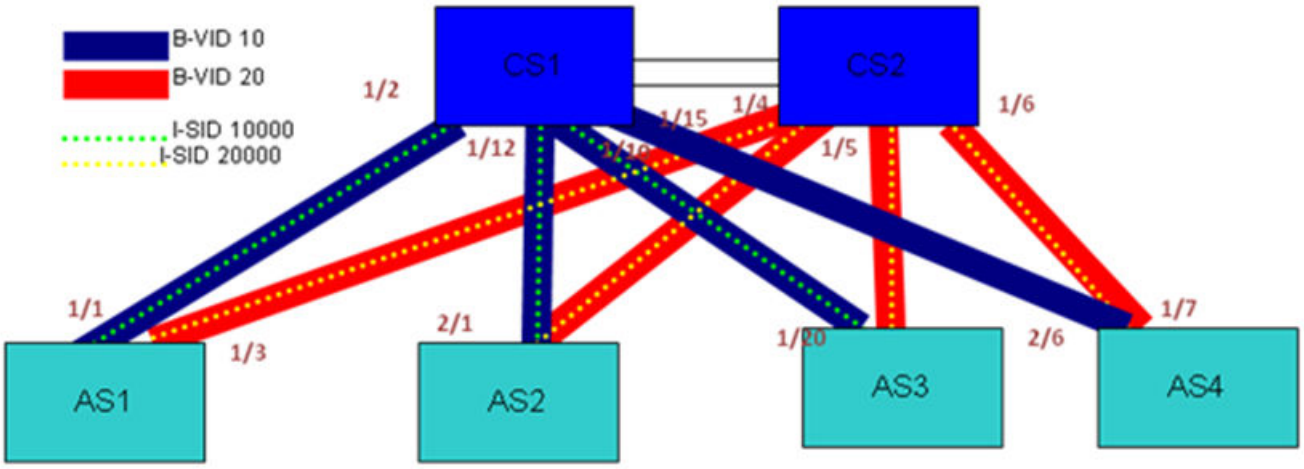
MMRP PDU Forwarding

- If MMRP is not enabled globally on the system, then the MMRP PDUs will be flooded in the hardware.
- If MMRP is enabled globally, then MMRP PDUs are captured to CPU for further processing.
  - In the LP, if MMRP is enabled on the source port of the PDU, then it is forwarded to MP for further processing.
  - If MMRP is not enabled on the source port of the PDU, then it is flooded to other ports from the LP.
- Flooding of MMRP PDUs occurs on the forwarding ports of the VLAN on which the PDU is received.

Sample topology

In the network shown in Figure 202 there are two E-LANS one for B-VLAN10 and ISID 10000 and other of B-VLAN20 and ISID 20000.

FIGURE 198 MMRP with PBB enabled



Sample MMRP configuration on AS1

```
device_AS1(config)#mmrp enable
device_AS1(config)#mmrp include-vlan 10 20
device_AS1(config)#int e 1/1 e 1/3
device_AS1(config-mif-1/1,1/3)#mmrp enable
device_AS1(config-mif-1/1,1/3)#mmrp include-vlan 10
```

Sample configuration on AS2

```
device_AS1(config)#mmrp enable
device_AS1(config)#mmrp include-vlan 10 20
device_AS1(config)#int e 2/1 e 2/3
device_AS1(config-mif-2/1,2/3)#mmrp enable
device_AS1(config-mif-2/1,2/3)#mmrp include-vlan 10
```

PBB configuration MLX for B-VLAN10

```
device_AS1(config)#vlan 10
device_AS1(config-vlan-10)#tag e 1/1
device_AS1(config)#
device_AS1(config)#router mpls
device_AS1(config-mpls)#vpls vinst 2000
device_AS1(config-mpls-vpls-vinst)#pbb
device_AS1(config-mpls-vpls-vinst-pbb)#vlan 10 isid 10000
device_AS1(config-mpls-vpls-vinst-vlan-10-isid-10000)#tagged ethernet 1/1
```

PBB configuration on Metro for B-VLAN10

```
device_AS1(config)#interface ethernet 1/1
device_AS1(config-if-e10000-1/1)#port-type backbone-network
device_AS1(config-if-e10000-1/1)#exit
device_AS1(config)#esi iptv-service encapsulation isid
device_AS1(config-esi-iptv-service)#isid 10000
device_AS1(config-esi-iptv-service-isid-10000)#exit
device_AS1(config)#esi iptv-carrier encapsulation bvlan
device_AS1(config-esi-iptv-carrier)#vlan 10
device_AS1(config-esi-iptv-carrier-vlan-10)#tagged ethernet 1/1
device_AS1(config-esi-iptv-carrier-vlan-10)#esi-client  iptv-service
```

Sample configuration on CS1

```
device_CS1(config)#mmrp enable
device_CS1(config)#mmrp include-vlan 10 20
device_CS1(config)#int e 1/2 e 1/10 e 1/12 e 1/15
device_CS1(config-mif-1/2, 1/10, 1/12, 1/15)#mmrp enable
```

PBB configuration MLX for B-VLAN 10

```
device_CS1(config)# vlan 10
device_CS1(config-vlan-10)#tag e 1/2 e 1/10 e 1/12 e 1/15
device_CS1(config)#
```

PBB configuration on Metro for B-VLAN 10

```
device_CS1(config)#interface e 1/2 e 1/10 e 1/12 e 1/15
device_CS1(config-if-e10000-1/1)#port-type backbone-network
device_CS1(config-if-e10000-1/1)#exit
device_CS1(config)#esi iptv-carrier encapsulation bvlan
device_CS1(config-esi-iptv-carrier)#vlan 10
device_CS1(config-esi-iptv-carrier-vlan-10)#tagged e 1/2 e 1/10 e 1/12 e 1/15
```

MMRP is used with PBB for the registration and declaration of Multicast B-DA MAC.

Declaration of MAC

The declaration of the multicast MAC is done by the BEB. In the topology described above, the declaration is sent if AS1 is

- Brocade NetIron CER Series and Brocade NetIron CES Series
  - when an ISID ESI IPTV-service is added as client to the B-VLAN ESI IPTV-carrier
  - when MMRP is enabled on the port belonging to a B-VLAN with ISID clients associated
  - when a port enabled with MMRP is tagged to B-VLAN with ISID clients associated
- Brocade NetIron MLX Series and Brocade NetIron XMR Series
  - when B-VLAN 10 ISID 10000 endpoint is created in the VPLS instance
  - when MMRP is enabled on the port where B-VLAN and ISID is configured

When MMRP sends Join PDU on ports in the B-VLAN. The attribute declared here would be the multicast flood MAC (011e.8300.2710).

Displaying the declared MAC on AS1

```
device_AS1# show mmrp attributes
Port        Vlan        Mac-address                    Registrar
Registrar                    Applicant
                                  State                  Mgmt                  State
-------------------------------------------------------------------------
1/1        10        011e.8300.2710          IN                            Fixed
Quiet Active
```

Registration of MAC

CS1 receives this declaration on port 1/2 and will register it. It will then propagate the declaration to all the ports of the B-VLAN that are in the forwarding state.

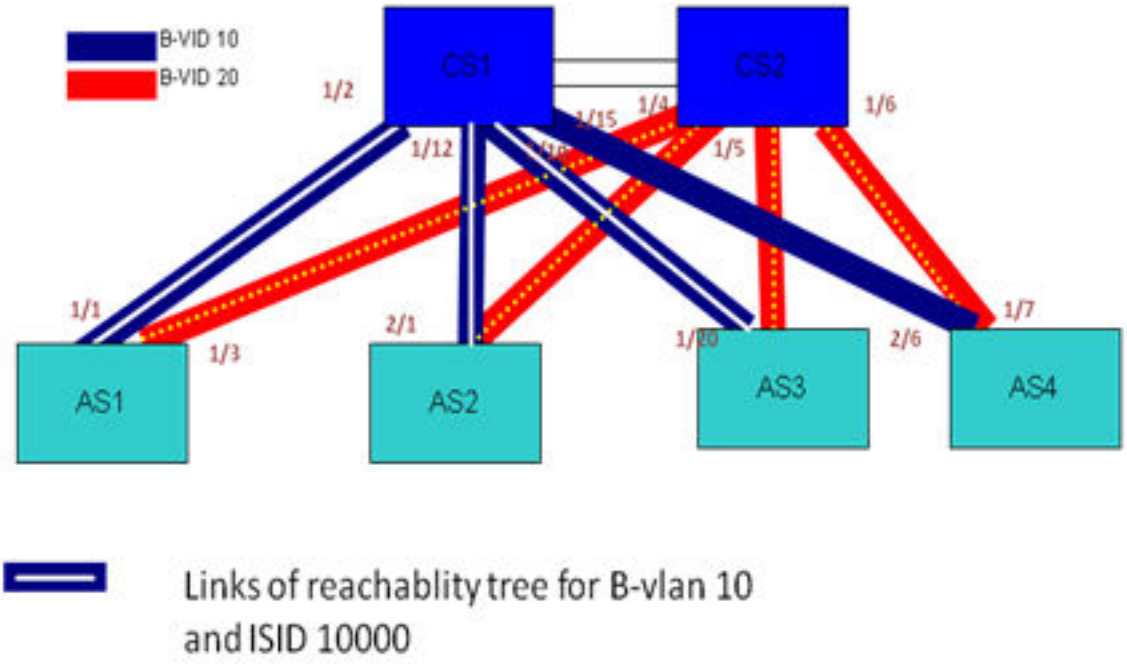
This dissemination will result in the registration of this MAC on AS2, AS3, and AS4 and reachability tree is formed. Similarly AS2, AS3 also declare for the flood MAC. AS4 will not declare because the I-SID is not terminated on AS4. CS1 now has ports connecting to AS1, AS2, and AS3 registered to the flood MAC for B-VLAN10. The Figure 203 shows the reachability tree for the B-VLAN10.

Displaying the Registered MAC on CS1

```
device_CS1#show mmrp attributes
Port        Vlan        Mac-address              Registrar
Registrar                       Applicant
                                  State                Mgmt                      State
-------------------------------------------------------------------------
```

```
1/2          10         011e.8300.2710      IN                                  Normal
Quiet Active
1/10         10          011e.8300.2710      IN                                   Normal
Quiet Active
1/12         10          011e.8300.2710      IN                                   Normal
Quiet Active
```

FIGURE 199 Reachability tree for B-VLAN 10 and ISID 10000



When packets with this MAC address reach the CS, it will multicast only on ports on which registration of the MAC address is present (for example: it will multicast to AS1, AS2 and AS3 but not to AS4).

Unknown Unicast Multicasting in the above Topology:

1. AS2 is trying to send to a customer MAC C.
2. AS2 will flood the packet towards on port 2/1 the PBB network with the B-SA as the MS2 (Base MAC of AS2) and B-DA as Multicast flood MAC, B-VLAN as 10 and ISID as 10000.
3. CS1 on receiving this packet will multicast to the ports on which it has registered the flood MAC, in this case 1/2 and 1/10 (it will not send on 1/12 because of source port suppression).
4. If AS1 has been learnt the customer MAC C then it will strip the PBB header do Layer 2 forwarding towards the customer.

Configuring MMRP

MMRP Operation Overview

MMRP defines an MRP application that provides the extended filtering services. MMRP makes use of the following:

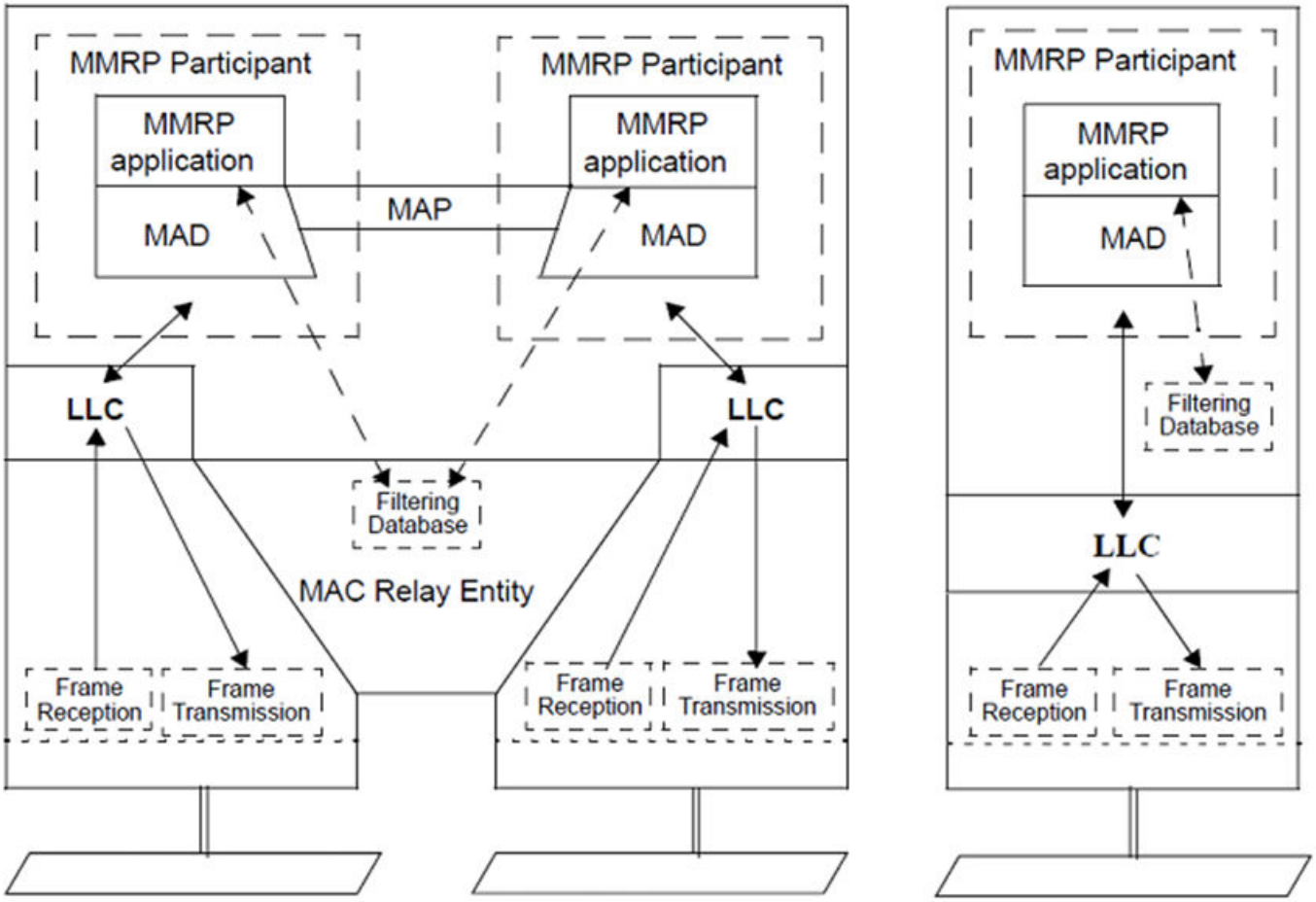
- The declaration and propagation services offered by MAD and MRP to declare and propagate Group membership, Group service requirement, and individual MAC address information within the LAN
- The registration services offered by MAD to allow Group membership, Group service requirement, and individual MAC address information to control the frame filtering behavior of participating devices.

Figure 204 illustrates the architecture of MMRP in the case of a two-Port Bridge and an end station, for a given VLAN Context. Where MMRP is used in multiple VLAN Contexts, an instance of the MMRP Participant exists for each VLAN context.

As shown in the diagram, the MMRP Participant consists of the following components:

- The MMRP application
- MRP Attribute Propagation
- MRP Attribute Declaration

FIGURE 200 MMRP components



Enabling MVRP at global level

MMRP must be enabled globally to allow the device to participate in the MMRP (IEEE 802.1ak) protocol.

device(config)# mmrp enable

Syntax: [no] mmrp enable

MMRP is disabled by default.

MMRP include-vlan configuration

The **mmrp include-vlan** command is used to configure a set of VLANs on which MMRP is allowed to participate. The global configuration is applicable to all the MMRP enabled ports unless an explicit configuration is made on the port.

device(config)# mmrp include-vlan 500 600 700

Syntax: [no] mmrp include-vlan *vlan id orrange*

By default, no VLANs are included. Only 256 B-VLANs are allowed to participate in MMRP. The range of values available is 1-4090.

On Brocade NetIron CER Series and Brocade NetIron CES Series devices, the VLAN should be in the B-VLAN ESI.

On Brocade NetIron MLX Series and Brocade NetIron XMR Series devices, the B-VLAN must be a layer 2 (L2) VLAN.

Global Timer Configuration

Use the **mmrp timer** command to configure the join, leave, and leave-all timers at global level for MMRP.

```
device(config)# mmrp timer join 400 leave 1400 leave-all 10000
```

Syntax: [no] mmrp timer join *value* **leave** *value* **leave-all** *value*

The **mmrp timer** command sets the timer value in milliseconds (ms). The default value for the Join timer is 200 ms. The allowable range for the Join timer is 200 to 100000000ms.

The default value for the **leave** timer is 1000ms. The allowable range for the Leave timer is 1000 to 100000000ms. The recommended value for the **leave** timer is 5000ms.

The default value for the **leave-all** timer is 10000ms. The allowable range for the **leave-all** timer is 10000 to 100000000ms.

Configuration restrictions

The Leave timer should be greater than or equal to twice the join timer plus 600ms.

Leave-all timer should be large relative to the Leave timer; recommended value is at least three times the value of Leave timer.

Per Interface configuration

To configure MMRP at the interface level, MMRP must be enabled at global level first.

Enabling MMRP on an interface

```
device(config-if-e1000-1/1)#mmrp enable
```

Syntax: [no] mmrp enable

By default, MMRP is disabled on all interfaces. After MMRP is enabled globally, use the **mmrp enable** command on the interfaces on which MMRP is required.

MMRP include-vlan configuration

Use the **mmrp include-vlan** command to configure a set of VLANs on which MMRP is allowed to participate on an interface. The interface VLANs are allowed only when they are configured under global **mmrp include-vlan** command.

```
device(config-if-e1000-1/1)# mmrp include-vlan 100 to 200
device(config-if-e1000-1/1)# mmrp include-vlan 500 600 700
```

Syntax: [no] mmrp include-vlan *vlan id orrange*

By default, if no port level configuration exists, then MMRP will operate on a globally configured include-vlan.

The acceptable values include the range of 1 to 4090. Only 256 B-VLANs are allowed to participate in MMRP.

MMRP interface level timers

Use the **mmrp timer** command to configure the join, leave, and leave-all timers at global level for MMRP.

```
device(config)# mmrp timer join 400 leave 1400 leave-all 10000
```

Syntax: [no] mmrp timer join *value* **leave** *value* **leave-all** *value*

The **mmrp timer** command sets the timer value in milliseconds (ms). The default value for the **join** timer is 200 ms. The allowable range for the Join timer is 200 to 100000000ms.

The default value for the **leave** timer is 1000ms. The allowable range for the Leave timer is 1000 to 100000000ms. The recommended value for the **leave** timer is 5000ms

The default value for the **leave-all** timer is 10000ms. The allowable range for the Leave-all timer is 10000 to 100000000ms.

Configuration restrictions

The Leave timer should be greater than or equal to twice the join timer plus 600ms.

Leave-all timer should be large relative to the Leave timer; recommended value is at least three times the value of Leave timer.

MMRP registration-mode configuration

The **mmrp registration-mode forbidden** command allows you to configure registration mode for MACs to be forbidden.

```
device(config-if-e1000-1/1)# mmrp registration-mode forbidden 011E.8300.3001
```

Syntax: [no] mmrp registration-mode forbidden vlan *vlan-id* **mac-address** [*macaddr* | *macaddr* **to** *macaddr*]

By default, registration mode for a MAC attribute is normal. When ISIDs are configured locally the registration mode for the MAC is fixed.

MMRP point-to-point configuration

Configuring an interface as point-to-point for MMRP.

Single interface example

```
device(config)# interface e 1/1
device(config-eth-1/1)# mmrp enable
device(config-eth-1/1)# mmrp include-vlan 300
device(config-eth-1/1)# mmrp include-vlan 1000 1100
device(config-eth-1/1)# mmrp timer join 400 leave 1200 leave-all 10000
device(config-eth-1/1)# mmrp point-to-point
```

Multiple Interface (consecutive) configuration example

```
device(config)# int e 1/1 to e 1/2
device(config-mif-1/1-1/2)# mmrp enable
device(config-mif-1/1-1/2)# mmrp include-vlan 300
device(config-mif-1/1-1/2)# mmrp include-vlan 1000 1100
device(config-mif-1/1-1/2)# mmrp timer join 400 leave 1400 leave-all 10000
device(config-mif-1/1-1/2)# mmrp point-to-point
```

Multiple Interface (non consecutive) configuration example

```
device(config)# int e 1/1 e 1/3 e 1/5
device(config-mif-1/1,1/3,1/5)# mmrp enable
device(config-mif-1/1,1/3,1/5)# mmrp include-vlan 300
device(config-mif-1/1,1/3,1/5)# mmrp include-vlan 1000, 1100
device(config-mif-1/1,1/3,1/5)# mmrp timer join 400 leave 1400 leave-all 10000
device(config-mif-1/1,1/3,1/5)# mmrp point-to-point
```

Syntax: [no] mmrp point-to-point

By default, point-to-point is disabled.

Syslog messages

The following syslog messages may occur when using this feature.

1. When MMRP is disabled globally.

   ```
   <14>Sep 13 15:36:48 3-1-A MMRP is disabled globally.
   ```

2. When a new MAC is registered.

   ```
   <14>Sep 13 15:36:48 3-1-A MMRP Mac 011e.8300.2710 registered on port 1/1 vlan 100
   ```

3. When an MMRP MAC is removed.

   ```
   <14>Sep 13 15:36:48 3-1-A MMRP Mac 011e.8300.2710 is removed from port 1/1 vlan 100
   ```

CLI Error Messages

The following Error messages are introduced for this feature

1. Timer configuration. The timer configuration in MMRP has the following two restrictions
   - – Leave timer should be greater than or equal to twice the join timer plus 600ms. If this condition is not satisfied then the following error message will be displayed.

     ```
     Error: leave timer value must be greater than twice the join timer plus 600ms
     ```

   - – Leave-all timer should be large relative to leave timer; recommended value is at least three times the value of leave timer.

     ```
     Error: leave-all timer value must be greater than three times the leave timer value
     ```

2. If MMRP is disabled globally any MMRP command is issued then the following error will be displayed.

   ```
   Error: MMRP is disabled globally, operation rejected.
   ```

3. If MMRP is enabled on non-existing vlan then following error will be thrown

   ```
   Error: Vlan <vlan id> not configured
   ```

4. If at the interface level if MMRP is enabled on the vlan is which is not present in the global configuration then following error message will be displayed.

Example

```
device#conf t
device#(config)#mmrp enable
device(config)# mmrp include-vlan 10
device(config)#int e 1/1
device(config-if-e1000-1/1)#mmrp include-vlan 300
Error - mmrp vlan 300  must be configured at global level first
```

Reverse Path Forwarding

A number of common types of denial-of-service (DoS) attacks, including Smurf and Tribe Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. Reverse Path Forwarding (RPF) is designed to prevent such a malicious user from spoofing a source IP address by checking that the source address specified for a packet is received from a network to which the device has access. Packets with invalid source addresses are not forwarded. Optionally, you can log packets that fail the RPF test.

RPF is supported for IPv6 packets. Differences in RPF support in IPv4 and IPv6 are noted within this chapter where necessary.

RPF configuration

Before you begin to configure Reverse Path Forwarding (RPF), review the following sections:

- Configuration considerations for RPF on page 823
- Special considerations for configuring RPF on Brocade NetIron CES Series and Brocade NetIron CER Series devices on page 824
- Special considerations for configuring RPF with ECMP routes on page 824
- RPF support for IP over MPLS routes on page 824
- RPF-compatible CAM profiles on page 824
- Configuring the global RPF command on page 825
- Enabling RPF on individual ports on page 825
- Configuring a timer interval for IPv6 session logging on page 826
- Suppressing RPF for packets with specified address prefixes on page 826

Configuration considerations for RPF

Consider the following points when you configure Reverse Path Forwarding:

- IP packets with a source IP address of 0.0.0.0 will always fail RPF check.
- If you attempt to enable the global RPF command on a system with incompatible CAM settings, the command will be rejected and you will receive a console message.
- Because the RPF feature requires that the entire IP route table is available in hardware, the feature must work in conjunction with Foundry Direct Routing (FDR). FDR is the default mode of operation for the device.
- You cannot configure RPF on a physical port that has VRF configured on it, or if the physical port belongs to a virtual interface with a VRF configuration.
- Only RPF loose mode is supported for GRE routes.
- If a default route is present on the router, loose mode will permit all traffic.
- RPF can only be configured at the physical port level. It should not be configured on virtual interfaces on the Brocade NetIron MLX Series and Brocade NetIron XMR Series.
- Brocade NetIron CER Series and Brocade NetIron CES Series devices provide support for uRPF for VE interfaces.
- IPv6 packets with a link-local source address are not subject to IPv6 RPF check.

- IPv6 RPF check is not supported for 6-to-4 tunnel routes.

Special considerations for configuring RPF on Brocade NetIron CES Series and Brocade NetIron CER Series devices

- Brocade NetIron CES Series and Brocade NetIron CER Series devices do not support IPv6.
- Unlike the Brocade NetIron XMR and MLX devices, port level granularity is not supported on Brocade NetIron CES Series and Brocade NetIron CER Series devices; therefore, RPF must be configured on the entire device either all in loose mode or all in strict mode.
- If the logging feature is enabled, it is enabled on the entire device. If the logging feature is disabled, logging for the entire device is disabled.
- You cannot configure RPF on a physical port that belongs to a virtual LAN (VLAN).
- If a combination of RPF, PBR, and ACL are configured on an interface, RPF takes precedence.
- The section Special considerations for configuring RPF with ECMP routes on page 824 does not apply to the Brocade NetIron CES Series and Brocade NetIron CER Series devices.

Special considerations for configuring RPF with ECMP routes

NOTE
This section applies only to the Brocade NetIron XMR Series and Brocade NetIron MLX Series devices.

RPF for IPv6 is not subject to the special considerations for configuring RPF with ECMP routes described in this section.

For a source IP address matching an ECMP route, RPF permits the packet if it arrives on any of the next-hop interfaces for that route. For example, if there are two best next hops for a network route 10.11.11.0/24, one pointing to 10.10.10.1 (Gigabit Ethernet 7/1) and the other to 10.10.30.1 (Gigabit Ethernet 7/12), then incoming packets with a source address matching 10.11.11.0/24 will be permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

A disadvantage of this configuration is that if some other route shares any of these next hops, the packets with a source IP address matching that route are also permitted from any of the interfaces associated with those next hops. For example, if 10.12.12.0/24 has the next hop 10.10.10.1, then packets from 10.12.12.0/24 are also permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

RPF support for IP over MPLS routes

For IPv4 routes over MPLS tunnels, the physical interface for an outgoing tunnel on which a route is assigned may not be the same as the one from which you receive packets. Consequently, only RPF loose mode is supported on MPLS uplinks. IPv6 is not currently supported over MPLS. When it is supported, it will only support RPF loose mode on MPLS uplinks.

RPF-compatible CAM profiles

NOTE
This section applies only to the Brocade NetIron XMR Series and Brocade NetIron MLX Series devices.

Not all CAM profiles are compatible with RPF. Table 105 lists all of the RPF-compatible CAM profiles by software release. Refer to "CAM partition profiles" for a description of each of the available CAM profiles.

TABLE 105 RPF compatible and non-compatible CAM profiles

Software release	Compatible CAM profiles	Non-compatible CAM profiles
XMR or MLX	default	ipv4-ipv6
	ipv4	ipv4-vpls
	ipv4-vpn	l2-metro
	ipv6	l2-metro-2
	mpls-l3vpn	mpls-vpls
	mpls-l3vpn-2	mpls-vpls-2
	ipv4-ipv6-2	mpls-vpn-vpls
	multi-service-2	multi-service
	multi-service-4	

Configuring the global RPF command

Before you can enable RPF to operate on a device, you must first configure RPF globally. There are separate commands for IPv4 and IPv6, as shown in the following examples.

> **NOTE**
> IPv6 configurations are not supported on Brocade NetIron CES Series and Brocade NetIron CER Series devices.

For IPv6 configurations, use the following command.

Syntax: ipv6 reverse-path-check

```
Brocade(config-if-e1000-1/4)# rpf-mode ?
  loose    Allow packets forwarding if there is a route to source
  strict   Allow packets forwarding if route to source is towards the incoming
           port
Brocade(config-if-e1000-1/4)# rpf-mode strict ?
  log   Log packets that fail RPF check and are to be dropped
Brocade(config-if-e1000-1/4)# rpf-mode strict
```

For IPv4 configurations, use the following command.

```
device(config)# reverse-path-check
```

Syntax: reverse-path-check

```
device(config)# ipv6 reverse-path-check
```

Enabling RPF on individual ports

After RPF has been configured globally for a device, it must be configured on every interface that you want it to operate. The RPF feature can be configured on physical Ethernet interfaces. There are two modes, "strict" and "loose," that can be configured to enforce RPF on IP addresses for packets arriving on a given interface:

- In **loose** mode, RPF permits a packet as long as the source address matches a known route entry in the routing table. It will drop a packet if it does not match a route entry. Note that if a default route is present, loose mode will permit all traffic.
- In **strict** mode, RPF requires that a packet matches a known route entry as described in loose mode and also that it arrives at the interface as described in the router table's next hop information. It will drop a packet that does not match both of these criteria.

Configuring RPF on a port requires separate commands for IPv4 and IPv6. To configure RPF on a port, use the IPv4 or IPv6 command, as shown in the following examples.

For IPv4 configurations, use the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# rpf-mode strict log
```

Syntax: [no] rpf-mode [**loose** | **strict**] [**log**]

For IPv6, use the following commands.

```
device(config)# interface ethernet 3/1
device(config-if-e1000-3/1)# rpf-mode-ipv6 strict log
```

Syntax: [no] rpf-mode-ipv6 [**loose** | **strict**] [**log**]

There are two modes in which you can enforce RPF on IP sources address for packets that arrive on a configured interface:

- **The loose** option configures RPF in the loose mode.
- The **strict** option configures RPF in the strict mode.

The **log** option directs RPF to log packets that fail the RPF test. Enabling RPF logging may lead to high CPU utilization on the interface module because packets that fail the RPF check test are dropped in software. Only syslog entries are created by this option. No SNMP traps are issued by this option.

The ACL or RPF logging mechanism on the interface modules log a maximum of 256 messages per minute, and send these messages to the management module. A rate-limiting mechanism has been added to rate-limit the number of messages from the interface module CPU to the management module CPU to 5 messages per second. Because this delays the delivery of messages to the management module, in the worst case scenario with all 256 packets arriving at the same time on the interface module, the time values stamped by the management module on the messages will vary by as much as 60 seconds.

Configuring a timer interval for IPv6 session logging

You can use the **ipv6 session-logging-age** command to globally configure a timer interval for IPv6 session logging. The timer interval is set for 3 minutes in the following example.

```
device(config)# ipv6 session-logging-age 3
```

Syntax: [no] ipv6 session-logging-age minutes

The *minutes* variable sets the timer interval for logging. Configurable values are from 1 through 10 minutes. The default value is 5 minutes.

You can use the **show log** command to view RPF messages, as shown in the following example.

```
device# show log
Dec 18 19:32:52:I:IPv6 RPF: Denied 1 packet(s) on port 1/2 tcp fec0:1::2(0) -> 4500:1::2(0)
```

Suppressing RPF for packets with specified address prefixes

> NOTE
> This section is not applicable for the Brocade NetIron CES Series and Brocade NetIron CER Series devices because, with these devices, RPF takes precedence over PBR and ACLs.

You can suppress RPF packet drops for a specified set of packets using inbound ACLs. To suppress RPF packets:

1. Create an IPv4 or IPv6 ACL that identifies the address range that you do not want dropped.
2. Specify the flag to the ACL permit clause of the **suppress-rpf-drop** command.

When a packet that fails the RPF check and matches the specified ACL permit clause with the suppress-rpf-drop flag set, it is forwarded as a normal packet and it is accounted as a "**unicast RPF suppressed drop packet,**" as described in Displaying RPF statistics on page 827.

NOTE
The **suppress-rpf-drop** command is not supported on Brocade NetIron CES Series and Brocade NetIron CER Series devices.

The following example demonstrates the configuration of the IPv4 ACL named **"access-list 135"** which permits traffic from the source network 10.4.4.0/24 even if the RPF check test fails.

```
device(config)# access-list 135 permit ip 10.4.4.0.0.0.0.255 any suppress-rpf-drop
device(config)# access-list 135 permit ip any any
```

The following example demonstrates the configuration of the IPv6 ACL named "**rpf1**" which permits traffic from the source host 2002::1 even if the RPF check test fails.

```
device(config)# ipv6 access-list rpf1
device(config-ipv6-access-list rpf1)# permit tcp host 2002::1 any suppress-rpf-drop
```

Syntax: suppress-rpf-drop

In the following example, the IPv4 ACL 135 is applied as an inbound filter on Ethernet interface 7/5.

```
device(config)# interface ethernet 7/5
device(config-if-e1000-7/5)# rpf-mode strict
device(config-if-e1000-3/1)# ip access-group 135 in
```

NOTE
If the physical port is a member of a virtual interface, the ACL will have to be applied to the virtual interface instead of the physical port.

Excluding packets that match the routers default route

The **urpf-exclude-default** and **ipv6 urpf-exclude-default** commands direct the Brocade router to drop packets whose source address matches the routers default route and increment the RPF drop counter. Using this feature requires that RPF be configured globally first. This feature is configured separately for IPv4 and IPv6 as described in the following examples.

For IPv4, use the following commands.

```
device(config)# reverse-path-check
device(config)# urpf-exclude-default
```

Syntax: urpf-exclude-default

For IPv6, use the following commands.

```
device(config)# ipv6 reverse-path-check
device(config)# ipv6 urpf-exclude-default
```

Syntax: ipv6 urpf-exclude-default

Displaying RPF statistics

To display information about RPF configuration and packets that have been dropped because they failed the RPF check, use the **show ip interface** or the **show ipv6 interface** command as shown.

For IPv4, use the following command.

```
device# show ip interface ethernet 7/1
Interface Ethernet 7/1 (384)
    port enabled
    port state: UP
    ip address: 10.2.3.4/8
    Port belongs to VRF: default
    encapsulation: Ethernet, mtu: 1500
    MAC Address 000c.db24.a6c0
    directed-broadcast-forwarding: disabled
    No inbound ip access-list is set
    No outbound ip access-list is set
    No Helper Addresses are configured
    RPF mode: strict RFP Log: Disabled
    376720 unicast RPF drop 36068 unicast RPF suppressed drop
```

For IPv6 configurations, use the following command.

```
device#show ipv6 interface ethernet 3/1
Interface Ethernet  3/1 is down, line protocol is down
  IPv6 is enabled, link-local address is
  Global unicast address(es):
  Joined group address(es):
    ff02::2
    ff02::1
  MTU is 1500 bytes
  ICMP redirects are disabled
  ND DAD is enabled, number of DAD attempts: 3
  ND reachable time is 30 seconds
  ND advertised reachable time is 0 seconds
  ND retransmit interval is 1 seconds
  ND advertised retransmit interval is 0 seconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  No Inbound Access List Set
  No Outbound Access List Set
  IPv6 RPF mode: Strict  IPv6 RPF Log: Enabled
  RxPkts:       0                      TxPkts:   0
  RxBytes:      0                      TxBytes:  0
  IPv6 unicast RPF drop: 0
  IPv6 unicast RPF suppressed drop: 0
```

NOTE

The RPF accounting information is always available through the physical interface, even if the physical port belongs to one or more VE's

Table 106 describes the RPF statistics displayed when using the **show ip interface** or **show ipv4 interface** command. They are displayed in **bold font** .

TABLE 106 RPF statistics by port

Field	Description
RPF mode:	This display parameter can have one of the following two values: • loose - RPF will permit a packet as long as the source address matches a known route entry in the routing table. It will drop a packet if it does not match a route entry. • strict - RPF requires that a packet matches a known route entry as described for loose mode and also that it arrives at the configured interface as described in the router table's next hop information. It will drop a packet that does not match both of these criteria.
RPF Log:	This display parameter displays the RPF Log Configuration Status: • Enabled - The RPF log feature has been configured.

TABLE 106 RPF statistics by port (continued)

Field	Description
	• Disabled - The RPF log feature has not been configured
number unicast RPF drop	The number of packets that have been dropped due to failure of the RPF test.
number unicast RPF suppressed drop	The number of packets that would have been dropped due to failure of the RPF test but were not dropped because they matched conditions set in an ACL with the flag set in the **suppress-rpf-drop** command.

Clearing RPF statistics for a specified IPv4 interface

To clear RPF statistics on a specific IPv4 physical interface, use the **clear ip interface ethernet** command.

Syntax: clear ip interface ethernet slot/port

Use the **ethernet** parameter to specify the Ethernet or port.

The *slot/port* variables specify the interface for which you want to clear RPF statistics.

Clearing RPF statistics for all IPv4 interfaces within a router

To clear RPF statistics on all IPv4 physical interfaces within a router, use the **clear ip interface counters** command.

Syntax: clear ip interface counters

Clearing RPF statistics for a specified IPv6 interface

To clear RPF statistics on a specific IPv6 physical interface, use the **clear ipv6 interface** command.

Syntax: clear ipv6 interface ethernet slot/port

Use the **ethernet** parameter to specify the Ethernet port.

The *slot/port* variables specify the interface for which you want to clear RPF statistics.

Clearing RPF statistics for all IPv6 interfaces within a router

To clear RPF statistics on all IPv6 physical interfaces within a router, use the **clear ipv6 interface counters** command.

Syntax: clear ipv6 interface counters

Displaying RPF logging

If you set the log option of the **rpf-mode** command, the packets are saved to the system log. To display the log, use the following command.

```
device# show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 1305 overruns)
    Buffer logging: level ACDMEINW, 50 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Dynamic Log Buffer (50 lines):
May 11 12:12:54:I:RPF: Denied 1 packets on port 7/5 tcp 10.4.4.1(0) -> 10.6.7.8(0)
```

NOTE

A maximum of 256 RPF log messages are logged per minute.

sFlow

sFlow is a system for collecting information about traffic flow patterns and quantities within and among a set of devices. You can configure a device to perform the following tasks:

- Sample packet flows
- Collect packet headers from sampled packets to gather ingress and egress information on these packets
- Compose flow sample messages from the collected information
- Relay messages to an external device known as a collector

Participating devices can also relay byte and packet counter data (counter samples) for ports to the collector.

The port connected to the collector forwards sFlow packets in management VRF and default VRF. The Brocade implementation of sFlow data collection supports AS path information in the following types of sFlow packets:

- Non-default VRF IPv4 sampled packets
- Non-default VRF IPv6 sampled packets
- Default VRF IPv4 sampled packets
- Default VRF IPv6 sampled packets

RFC 3176, "InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks" describes sFlow. Refer to this RFC to determine the contents of the sampled packet.

Brocade supports sFlow v5, which replaces the version outlined in RFC 3176.

sFlow event workflow

If the sFlow destination is IPv6 and the sFlow Agent IPv6, then an IPv6 agent will be selected from the configured interface. Otherwise, IPv4 will be selected from the configured interface ID from the **sFlow agent** command. If the sFlow agent is not configured, the router ID is used.

If the sFlow destination is IPv4, and the sFlow agent is configured, then an IPv4 agent will be selected from the configured interface. If the sFlow agent is not configured, the router-ID is used.

The Agent IP address selects the first IP address in the interface IP address list. The Agent IPv6 address is unspecified by default. Use the **show sflow** command to verify the interface IP address list.

The status of an IP-port (UP, DOWN) will not impact the sFlow source IP.

The adding or deleting of IP addresses on the interface upon which the sFlow agent interface is configured or a router ID change will trigger the following events:

1. Router ID event:

   If the sFlow agent is not configured, or has been configured but an IP interface does not contain an IP address, then the sFlow agent will use the current management VRF router ID (if any). If the management VRF has changed, then the sFlow agent will

also update the agent IP address. However, if the management VRF is disabled or assigned to the default VRF (default behavior), then a router ID event will be applied for the global router ID. The sFlow agent will be updated accordingly.

2. Adding IP address event:

   Adding an IP address on an interface upon which the sFlow agent is configured on will impact an agent-IP based on the following scenarios:

   - If this IP address is the first IP address in the table then the sFlow agent selects it.
   - If the added IP address is positioned on the top of the IP table (due to IP address sequence order), then an agent IP will be reassigned to it. However, if it is not, then it will not impact the agent IP address.

3. Deleting IP address event:

   Deleting an IP address on an interface that the sFlow agent is configured on will impact an agent-IP based on the following scenarios:

   - If the deleted IP address is an equivalent to agent IP address then the next IP address on the same interface will be selected.
   - If no more IP addresses are found on that interface, then the agent IP address will use the router ID as the default behavior. sFlow agent IPv6 will be unspecified. Otherwise there is no action.

Configuration considerations

- Sample data is collected from inbound traffic on ports enabled for sFlow, but it does not collect the outbound traffic, even if the sFlow forwarding is enabled in the egress port. However, byte and packet counters that are sent to the collector include ingress and egress traffic statistics. The actual IP source address of the IP header is taken from the router port address of the best route to the sFlow collector IP address.
- Interface module processors directly forward sFlow packets to the specified sFlow collector. The sFlow collector is reachable by the way of ports on any of the Interface modules. Brocade requires sFlow collector to be connected to non-management port.
- For multicast traffic, sFlow sampling will display incorrect output for egress VLANs. In some configuration scenarios ingress VLAN may be incorrect.
- sflow is implemented in the default VRF only. Therefore, sflow data is only accessible by the sflow collector (sflow destination hosts) defined in the default VRF.

Source address

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the IP address of the device that sent the data.

If the sFlow destination is IPv6 and the sFlow Agent IPv6, then an IPv6 agent will be selected from the configured interface. Otherwise, IPv4 will be selected from the configured interface ID from the sFlow agent command. If the sFlow agent is not configured, the router-ID is used.

If the sFlow Destination is IPv4, and the sFlow agent is configured, then an IPv4 agent will be selected from the configured interface. If the sFlow agent is not configured, the router-ID is used.

sFlow looks for an IP address in the following order, and uses the first address found:

- The router ID configured by the **ip router-id** command, in the interface IP address list. The Agent IPv6-address is unspecified by default. Use the **show sflow** command to verify the interface IP address list.
- The first IP address on the lowest-numbered loopback interface
- The first IP address on the lowest-numbered virtual interface

- The first IP address on any interface

NOTE
If an IP address is not already configured when you enable sFlow, the feature uses the source address 0.0.0.0. To display the agent_address, enable sFlow, and then enter the **show sflow** command. Refer to sFlow forwarding on page 839 and Displaying sFlow information on page 842.

NOTE
If you change the agent_address, you must disable and then re-enable sFlow to use the newly configured 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. Device ports send only the sampled traffic to the CPU. sFlow sampling requires high LP CPU usage, which can affect performance in some configurations, especially if a high sampling rate is implemented.

Configured rate and actual rate

When you enter a sampling rate value, this value is the configured rate. The software rounds the value you enter to the next higher power of 2 to obtain the actual rate. This value becomes the actual sampling rate. For example, if the configured sampling rate is 1000, then the actual rate is 1024; and the hardware samples 1 in 1024 packets. If the configured sampling rate is 1025, then the actual rate is 2048.

NOTE
This behavior applies to the Brocade NetIron XMR Series and Brocade NetIron MLX Series platforms and does not apply to the Brocade NetIron CES Series and Brocade NetIron CER Series devices. In Brocade NetIron CES Series and Brocade NetIron CER Series devices, the system does not apply rounding.

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.

The prefix length of IPv4 source and destination IP addresses is collected only if you configure BGP 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 number
- The route's source peer AS number
- The AS path to the destination

In BGP-configured routers, AS Path information is collected from each node traversed by the sFlow packets.

NOTE
AS communities and local preferences are not included in the sampled packets.

To obtain extended gateway information, use "struct extended_gateway" as described in RFC 3176

sFlow null0 sampling

This feature allows Brocade devices to sample null0 dropped packets. This is useful in cases such as DOS attack on a particular route.

Configuring steps

1. Enable sFlow.
2. Enable null0 sampling.
3. Configure null0 routes.

NOTE
Above commands can be performed in any order.

Feature characteristics

- IPv4, IPv4-VPN, IPv6 null0 routes can be sFlow sampled.
- Only explicitly configured null0 routes can be sFlow sampled. Implicit null0 drops cannot be sFlow sampled.
- By default, null0 sFlow sampling feature is disabled.

Limitations

- When this feature is enabled, due to sampling of more packets (discarded packets) than the usual number till now, the actual sampling rate for regular streams will be reduced.
- This feature does not support PBR related null0 drops.
- This feature does not support default null0 route drops.

Backward compatibility

The current sFlow functionalities and ACL based sFlow functionalities will co-exist with this feature. As the dropped packets hit TM, if mirroring is enabled on that port, these dropped packets will also get mirrored.

Enabling/disabling the null0 sFlow sampling

These commands include the enabling and disabling of the null0 sampling.

Enter the following command to enable sFlow sampling for null0 routes.

```
Brocade(config)#sflow null0-sampling
```

To disable null0 sampling, enter the following command.

```
Brocade(config)#no sflow null0-sampling
```

Syntax: [no] sflow null0-sampling

Configuring a null0 route

For configuring a route for null0 sampling, use the following command.

```
Brocade(config)#ip route 10.10.10.100/32 null0
```

Syntax: [no] [**ip** | **ipv6**] **route** *ip-addr* **null0**

Displaying sFlow show command

This command will display the configuration for sFlow.

```
Brocade(config)#show sflow
sFlow services are enabled.
```

```
sFlow management VRF is enabled.
sFlow management VRF name is default-vrf.
sFlow agent IP address: 55.55.55.56
sFlow agent IPV6 address: unspecified
sFlow source IP address: unspecified, UDP 8888
sFlow source IPv6 address: unspecified, UDP 8888
Collector IP 77.7.7.2, UDP 6343
Polling interval is 20 seconds.
Configured default sampling rate: 1 per 2048 packets.
0 UDP packets exported
124 sFlow samples collected.
133 sFlow management-vrf UDP packets dropped
0 ACL sFlow samples collected.
sFlow ports   Global Sample Rate   Port Sample Rate   Hardware Sample Rate
        1/5                2048               2048   port_down
        1/8                2048               2048   2048
sFlow Null-0 Sampling is Enabled.
```

Configuring sFlow statistics

When traffic is received in the sFlow enabled interface, packets are sent to the LP CPU. The packets are processed by sFlow module by adding sFlow header along with the packet header and thereafter sent to the sFlow collector. The statistics of sFlow samples are maintained in the sFlow collector.

Use this command in the sFlow module to display the total count per interface for both sFlow and ACL based samples in all the slots where sFlow is configured.

```
Brocade(config)# show sflow statistics
Sflow Ports                Flow Samples count     Acl Samples Count
1/1                   800                          0
1/5                   0                          900
2/1                   600                          0
Brocade(config)# show sflow statistics  ethernet 1/1
Sflow Ports                Flow Samples count     Acl Samples Count
1/1                   800                          0
```

Syntax: show sflow statistics

clear statistics sflow command clears all the statistics collected per interface.

```
Brocade(config)# clear statistics sflow
Brocade(config)# show sflow statistics  ethernet 1/1
Sflow Ports                Flow Samples count     Acl Samples Count
1/1                   0                          0
```

sFlow support for MPLS

In addition to the Layer 2 or Layer 3 information typically exported across devices, when sFlow sampling is configured on VPN endpoint interfaces, you can export MPLS or VPN information, such as VLL, VPLS, and VRF customer endpoint interfaces details. This functionality allows service providers to collect sFlow information from VPN customers.

For incoming packets to an endpoint interface sampled by sFlow, the following additional information is collected and exported in the sFlow packets:

- **MPLS VC information:** including the VC name, VC index, and VC label COS
- **MPLS tunnel information:** including the LSP tunnel name, the tunnel index as assigned by the router, and the tunnel COS used

  NOTE
  IP over MPLS (non-Layer 3 VPN or VRF) packets are not supported for sFlow processing.

sFlow with VPLS local switching

This feature allows sFlow to carry the original VLAN ID of the incoming traffic in scenarios where a VPLS instance has multiple endpoints and different endpoints with different VLAN IDs -- implementing automatic VLAN ID translation.

When VPLS CPU protection is enabled in conjunction with sFlow, hardware flooded with sFlow, hardware flooded with broadcast, multicast, and unknown unicast, packets are marked with a source VLAN ID of 0. The destination VLAN ID cannot be determined in such cases. This behavior applies to all VPLS traffic.

> NOTE
> You must configure MAs with different MD levels to monitor the different endpoints with different VLAN IDs in the same VPLS instance.

Configuring and enabling sFlow

To configure sFlow, you must specify the collector information. The collector is the external device to which you are exporting the sFlow data. Optionally, you can change the polling interval and the sampling rate. Next, you enable sFlow globally and then enable forwarding on individual interfaces.

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.

```
device(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* variable specifies the collector's IP address.

The *dest-udp-port* variable specifies the UDP port on which the sFlow collector will be listening for exported sFlow data. The default port number is 6343.

The sampled sFlow data sent to the collectors includes an agent_address field. This field identifies the device that sent the data. Refer to Source address on page 832.

Changing the polling interval

The polling interval defines how often sFlow byte and packet counter data for a port are sent to the sFlow collectors.

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.

```
device(config)# sflow polling-interval 30
```

Syntax: [no] sflow polling-interval secs

The *secs* variable 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 received 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 1024, on average, 1 in every 1024 packets forwarded on an interface is sampled.

You can change the default (global) sampling rate. You also can change the rate on an individual port, overriding the default sampling rate.

> NOTE
> sFlow uses CPU resources to send sFlow samples to the collector. If you set a low sampling value on a high rate interface (for example 10 GbE), the interface module CPU utilization can become high.

Configuration considerations

The sampling rate is a fraction in the form 1/N, meaning that, on average, one out of every N packets will be sampled. The **sflow sample** command at the global level or port level specifies N, the denominator of the fraction. A higher denominator means a lower sampling rate because fewer packets are sampled. Likewise, a lower number for the denominator means a higher sampling rate because more packets are sampled. For example, if you change the denominator from 2,000 to 512, the sampling rate increases because four times as many packets will be sampled.

> NOTE
> It is recommended that you do not change the denominator to a value lower than the default. Sampling requires CPU resources. Using a low denominator for the sampling rate can cause high CPU utilization.

Changing the 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 a sampling rate. For example, if you enable sFlow on ports 1/1, 1/2, and 5/1 and you configure the sampling rate on port 1/1 but leave the other two ports using the default rate, then changing the global sampling rate would apply to ports 1/2 and 5/1 but not port 1/1. sFlow uses the sampling rate you explicitly configured on the individual port even if you globally changed 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 retains the sampling rate it had when you disabled sFlow forwarding on the port, unless the sflow sampling rate is removed or moved to default rate.

Sflow sampling on Brocade NetIron CES Series and Brocade NetIron CER Series devices

> NOTE
> Sflow samples outbound traffic if the sflow enabled port is monitored by a mirror port.

On Brocade NetIron CES Series and Brocade NetIron CER Series devices, if mirrored Sflow packets are received in the LP CPU there is no option to distinguish them from regular Sflow packets.

Changing the default sampling rate

> NOTE
> The Brocade NetIron CES Series and the Brocade NetIron CER Series devices support sFlow sampling rate configuration on a per-port basis. The Brocade NetIron XMR Series and Brocade NetIron MLX Series devices support sFlow sampling rate configuration on a per-packet processor basis.

To change the default (global) sampling rate, enter a command such as the following at the global configuration level.

```
device(config)# sflow sample 1024
```

Syntax: [no] sflow sample *num*

The *num* variable specifies the average number of packets from which each sample will be taken. The sampling rate you configure is the actual sampling rate. You can enter a value from 512 through 1048576. 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 may want to configure the Gigabit Ethernet ports to use a higher sampling rate (gathering 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.

```
device(config-if-e10000-1/1)# sflow sample 8192
```

Syntax: [no] sflow sample num

The *num* variable specifies the average number of packets from which each sample will be taken. The software rounds the value you enter up to the next odd power of 2. The actual sampling rate becomes one of the values listed in Changing the default sampling rate on page 837.

Configuring the sFlow source interface

sFlow source interface is globally defined for all sFlow destinations. For detailed information about the sFlow agent, refer to sFlow event workflow on page 831.

```
device(config)#sflow source [ipv6] [[ethernet | loopback | ve | pos <interface-id>] |[null0]] [<udp-port-
id>]
```

Syntax: [no] sflow source [**ipv6**] [[**ethernet** | **loopback** | **ve** | **pos** *interface-id*] | [**null0**]] [*udp-port-id*]

By default, the sFlow source interface is not specified, and the outgoing interface of an sFlow packet will be used as the source interface and address. The sFlow source port is 8888 by default.

Use the IPv6 option parameter indicate the IPv6 sFlow source address. If the destination IPv6 type does not match with the sFlow source IP address then the default behavior will be taken.

The sFlow source UDP for IPv4 is independent of IPv6.

The Null0 option is used to drop the sFlow sample with this source, while maintaining sFlow statistics.

Configuring the sFlow agent interface

The sFlow agent interface is globally defined for all sFlow destinations.

Configuration considerations

- By default, the sFlow agent is not specified, and the sFlow datagram will use the router ID as the agent ID.
- The user has the ability to configure the sFlow agent interface for IPv4 and IPv6.

```
device(config)#sflow agent [ipv6] [[ethernet | loopback | ve | pos <interface-id>]
```

Syntax: [no] sflow agent [**ipv6**] [[**ethernet** | **loopback** | **ve** | **pos** *interface-id*]

The **ipv6** keyword will indicate IP version 6 from the configured interface ID. The optional keyword will be followed by the interface type.

The command**no sflow agent** command with the specific parameters removes the specified agent interface and reassigns the agent-IP to the router-ID as in the default behavior.

Configuring the sFlow management VRF

The **sflow management-vrf-disable** command is used to disable the management VRF for sFlow and using the default VRF instance. By default, the management VRF is enabled on sFlow.

```
device(config)#sflow management-vrf-disable
```

Syntax: [no] sflow [**management-vrf-disable**]

The **no sflow management-vrf-disable** command disables the use of management VRF on sFlow and enables the default VRF instance.

> NOTE
> The output of the **show running-config** command does not show "management-vrf-disable" because it is the default behavior. If the **no sflow management-vrf-disable** command has been used, "management-vrf-disable" will appear in the output to the show running-config command.

sFlow forwarding

sFlow exports data only for the interfaces on which you enable sFlow forwarding. You can enable sFlow forwarding on Ethernet or POS interfaces.

> NOTE
> sFlow forwarding enables sampling of data packets received on sFlow-enabled ports and does not sample data packets that leave sFlow-enabled ports.

To enable sFlow forwarding:

- Globally enable the sFlow feature.
- Enable sFlow forwarding on individual interfaces.

> NOTE
> Before you enable sFlow, make sure the device has an IP address that sFlow can use as its source address. Refer to Source address on page 832 for the source address requirements.

Enabling sFlow forwarding

To enable sFlow forwarding, enter commands such as the following.

```
device(config)# sflow enable
device(config)# interface ethernet 1/1 to 1/8
device(config-mif-1/1-1/8)# sflow forwarding
```

These commands globally enable sFlow, then enable sFlow forwarding on Ethernet ports 1/1 through 1/8. You must use both the **sflow enable** and **sflow forwarding** commands to enable the feature.

Syntax: [no] sflow enable

Syntax: [no] sflow forwarding

NOTE
Data for POS ports is sampled using Ethernet format. The PPP or HDLC header of the sampled POS packet is replaced with an Ethernet header. PPP or HDLC control packets or IS-IS packets transmitted or received at a POS port are not sampled. Such packets are not included in the number of packets from which each sample is taken.

NOTE
sFlow packets cannot be forwarded from a management interface. You must configure an IP interface on an Interface module to forward sFlow packets.

NOTE
Configuring sFlow with Provider Bridge (PB) or Provider Backbone Bridges (PBB) port-type is not supported on the Brocade NetIron CES Series and Brocade NetIron CER devices.

ACL-based Inbound sFlow

Multi-Service IronWare software supports using an IPv4 or IPv6 ACL to select sample traffic to be sent to an sFlow collector. The data matching an ACL clause can be collected to observe traffic flow patterns and quantities between a set of switches and routers. To accommodate collecting sFlow through standard procedures and using ACL-filtered traffic, the proprietary Tag Type 1991 encapsulates the sFlow samples obtained through ACL-based sFlow and separates them from the sequence flow of other sFlow samples. Figure 205 shows the format of an sFlow packet, which illustrates the differences between a standard sFlow payload and an ACL-based payload.

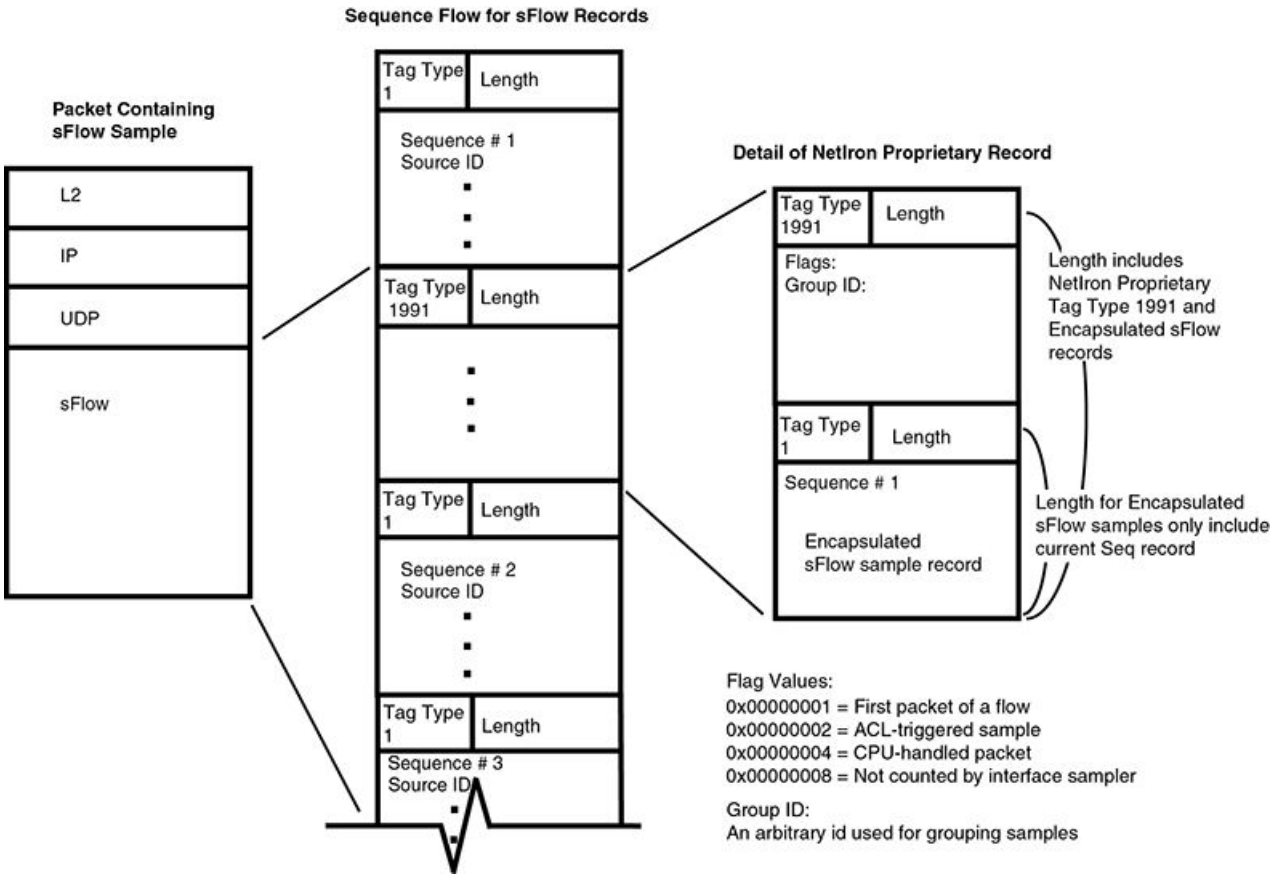
Figure 205 shows sFlow in a UDP packet. Within the UDP packet, the sFlow contents are carried in individual samples that are identified by a Tag Type and a Length variable. The standard values for the Tag Types are 1 (sampled packet) and 2 (counter sample). The Length variable describes the length of the sample. Within the sample are other variables including the Sequence number and the Source ID.

Brocade has introduced the proprietary Tag Type 1991 to identify ACL-based sFlow samples. For these samples, standard Tag Type 1 samples collected using ACL-based Inbound sFlow are encapsulated in a Tag Type 1991 sample. The Length variable identifies the entire length of the Tag Type 1991 sample including the encapsulated Tag Type 1 sample.The encapsulated sample has a Length variable of its own that only identifies the length of that sample.

The Tag Type 1991 samples are sequenced separately from the unencapsulated Tag Type 1 samples. For instance, in the packet detail described in "Sequence Flow for sFlow Records" in Figure 205, the top sFlow record with Tag Type 1 begins with the sequence number 1. The next sFlow record is Tag Type 1991, which indicates that the sample contained is from ACL-based sFlow. Encapsulated within this ACL-based sFlow sample is an sFlow sample record of Tag Type 1. The ACL-based sFlow sample (which contains the Tag Type 1 sample) is followed by an unencapsulated Tag Type 1 sFlow sample. That unencapsulated Tag Type 1 sFlow sample follows the sequence numbering of the first unencapsulated Tag Type 1 sFlow sample, which gives it a sequence number of 2.

This is useful in cases where an sFlow collector does not recognize Tag Type 1991. In these situations, the Tag Type 1991 samples can be ignored without disrupting the sFlow sequence numbers. It is also useful for identifying samples obtained using ACL-based sFlow on which other processing might be performed.

FIGURE 201 sFlow packet format



Configuring ACL-based Inbound sFlow

The following sections describe how to configure ACL-based Inbound sFlow:

- Configuration considerations for ACL-based Inbound sFlow on page 841
- Creating an ACL with an sFlow clause on page 842
- Displaying sFlow information on page 842

Configuration considerations for ACL-based Inbound sFlow

The following section describes configuration considerations for ACL-based Inbound sFlow:

- sFlow must be enabled on the router.
- **ACL-based mirroring:** The **mirror** and **copy-sflow** keywords are mutually exclusive on a per-ACL clause basis.
- **Port-based monitoring:** Port-based monitoring and ACL-based sFlow can co-exist on the same interface.
- **Port-based sFlow:** Port-based and ACL-based sFlow can co-exist on the same interface. When both features are configured on an interface, packets that qualify as ACL-based sFlow packets are sent to the collector as ACL sample packets. Also, the user can configure ACL-based sFlow on an interface without configuring port-based sFlow.
- **IP Receive ACLs:** IP Receive ACLs are used for filtering or rate-limiting management traffic. The **copy-sflow** keyword is also supported for IP Receive ACLs.

- **Policy Based Routing:** The **copy-sflow** keyword is applicable for PBR ACLs.
- **IPv4 ACL-based Rate Limiting:** When the **copy-sflow** keyword is used in an IPv4 Rate Limiting ACL, only traffic permitted by the Rate Limiting engine is copied to the CPU for forwarding to the sFlow collector.
- **IPv4 ACLs on VRF endpoints:** You can apply ACL-based sFlow for VRF endpoints; however, such packets are treated as regular sampled sFlow packets and do not carry proprietary encapsulation. This can create a minor skew of statistics projection.
- **Layer 2 ACLs:** The **copy-sflow** keyword is not supported for Layer 2 ACLs.
- If the **copy-sflow** keyword is used for a clause that is applied to the outbound direction, it is ignored.

Creating an ACL with an sFlow clause

The **copy-sflow** keyword has been added for inclusion in IPv4 and IPv6 ACL clauses to direct traffic that meets the criteria in the clause to be sent to the sFlow collector. In the following example, the ACL is used to direct syn-ack packets sent from a server at address 10.10.10.1.

```
access-list 151 permit tcp host 10.10.10.1 any established syn copy-sflow
access-list 151 permit any any
```

The **copy-sflow** keyword directs selected traffic to the sFlow collector. Traffic can only be selected using the **permit** clause.

You must apply the ACL to an interface using the**ip access-group command** as shown in the following example.

```
device(config)# int eth 1/1
device(config-if-e10000-1/1)# ip access-group 151 in
```

Displaying sFlow information

To display sFlow configuration information and statistics, enter the following command at any level of the CLI.

```
device(config)# show sflow
sFlow services are enabled.
sFlow management VRF is enabled.
sFlow management VRF name is blue.
sFlow agent IP address: 10.25.120.1
sFlow agent IPV6 address: unspecified
sFlow source IP address: unspecified, UDP 9999
sFlow source IPv6 address: 22::32, UDP 5544
2 collector destinations configured:
Collector IP 10.25.120.10, UDP 6343
Collector IPV6 10::32, UDP 6343
Polling interval is 20 seconds.
Configured default sampling rate: 1 per 2048 packets.
352 UDP packets exported
1 sFlow samples collected.
0 sFlow management-vrf UDP packets dropped
0 ACL sFlow samples collected.
sFlow ports Global Sample Rate Port Sample Rate Hardware Sample Rate
1/4 2048 10000 32768
```

Syntax: show sflow

Table 107 shows the output information provided by the **show sflow** command.

TABLE 107 sFlow information

Field	Description
sFlow services	The feature state, which can be one of the following: • disabled • enabled

TABLE 107 sFlow information (continued)

Field	Description
sFlow management VRF	Indication that sFlow is enabled to use the management VRF. Disabled means that sFlow is using the non-management VRF instance.
sFlow management VRF name	Management VRF name, if the management VRF is enabled on sFlow.
sFlow agent IP address	The IP address that sFlow is using in the agent_address field of packets sent to the collectors. Refer to Source address on page 832.
sFlow agent IPV6 address	The sFlow agent IPv6 address is unspecified by default. If configured, it will correspond to the IP address on the configured interface.
sFlow source IP address	The sFlow source IP address corresponds to the IP address on the configured interface. If an IP address is not configured on the interface, then it will be unspecified. However, if the source interface is null0, then it will be null0 on the interface.
sFlow source IPv6 address	The sFlow source IPv6 address that corresponds to the IP address on the configured interface. If an IP address is not configured, then the interface will be unspecified. However, if the interface is null0, then the configured interface will be null0.
UDP	The sFlow source UDP port default is 8888.
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.
UDP packets exported	The number of sFlow export packets the device has sent. NOTE Each UDP packet can contain multiple samples.
sFlow samples collected	The number of sampled packets that have been sent to the collectors.
sFlow ports	The ports on which you enabled sFlow.
Global Sample Rate	The global sampling rate for the device.
Port Sampling Rate	The sampling rates of a port on which sFlow is enabled.
Hardware Sample Rate	The actual sampling rate. This is the same as the Global Sample Rate

Displaying ACL-based sFlow statistics

Use the **show sflow** command to display the number of sFlow samples collected for ACL-based sFlow. These statistics are shown in bold in the following display.

```
device# show sflow
sFlow services are disabled.
sFlow agent IP address: 10.10.10.254
Collector IP 10.10.10.1, UDP 6343
Polling interval is 30 seconds.
Configured default sampling rate: 1 per 1024 packets.
0 UDP packets exported
0 sFlow samples collected.
5 ACL sFlow samples collected
sFlow ports   Global Sample Rate   Port Sample Rate   Hardware Sample Rate
        4/1                1024               8192   8192
```

Viewing BGP AS path sFlow statistics

The output of the **show sflow** command displays sFlow configuration information, and the elapsed time after the last sampling used for the BGP AS path table, and the interval for cleaning up the AS path table. The following example output shows that the AS path table has not been sampled within the last 51,385 seconds and that 3600 seconds, which is the default value, is the configured clean up interval.

```
Brocade(config)#show sflow
Slot 1 1/1 is disabled for sflow with sample rate = 2048 (actual rate = 2048)
Slot 1 1/2 is disabled for sflow with sample rate = 2048 (actual rate = 2048)
...
Slot 1 1/19 is disabled for sflow with sample rate = 2048 (actual rate = 2048)
Slot 1 1/20 is disabled for sflow with sample rate = 2048 (actual rate = 2048)
sflow destinations :
Total sflow sampling time = 0 (0)
Total sflow UDP time = 0 (0)
Total sflow ppcr tx time = 0 (0)
No sflow sampling on AS Path for 51385 sec
Sflow as path clean up wait interval 3600 sec
```

Clearing sFlow statistics

To clear the UDP packet and sFlow sample counters, use the **clear statistics** command.

```
device(config)# clear statistics
```

Syntax: clear statistics [**sflow**]

The **sflow** option clears the following values:

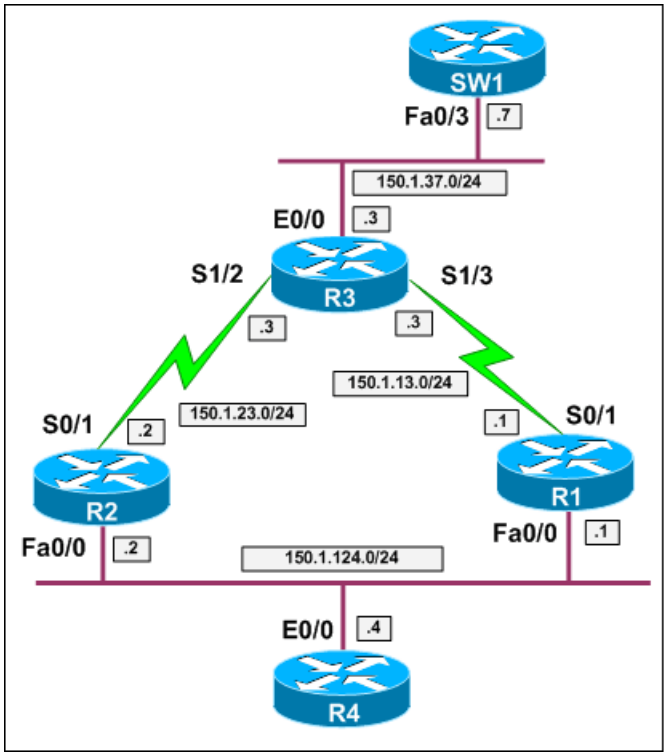
- UDP packets exported
- sFlow samples collected
- sFlow UDP packets dropped
- ACL sFlow samples collected

VLAN information in an sFlow packet

In Asymmetric routing scenarios, when sFlow is enabled, there is a possibility of a VLAN membership check failing if the reported interfaces do not have common VLAN membership. This will result in the reporting of incorrect VLAN information in the sFlow packet. The **routing-source-vlan-by-source-mac** command uses a mechanism provides the correct VLAN information.

An example of an asymmetric routing scenario is shown below. When sFlow is enabled on Router R3, if a request packet takes the path of R4->R2->R3->SW1, and R3 has its outgoing interface towards R4 as R3->R1->R4, there is a possibility of the VLAN membership check failing if the reported interface is S1/3 as S1/2 and S1/3 do not have common VLAN membership. This will result in reporting of the incorrect VLAN information in the sFlow packet. The routing-source-vlan-by-source-mac command will use a mechanism that will provide the correct VLAN information.

FIGURE 202 Example of an asymmetric routing scenario



Example of the routing-source-vlan-by-source-mac command

Syntax: [no] sflow routing-source-vlan-by-source-mac

The following is an example for routing-source-vlan-by-source-mac configuration.

```
(Device-config)#sflow routing-source-vlan-by-source-mac
```

The following command requires a reboot for it to be applicable.

Syntax: [no] system-max sflow-mac-cache <value>

For ageing of the cache entries:

Syntax: [no] sflow-mac-age-time <min>

To display the cache entries:

Syntax: show sflow-cache mac

To clear the sflow-cache entries:

Syntax: clear sflow-cache mac

Limitations

- For sFlow packets that are sampled for RPF Failure cases, the MAC entry found corresponding to the first VLAN is retrieved. Therefore, in the case of the port being a member of multiple VLANs and if the same source MAC is known against many of those VLANs, the exact source VLAN may not be returned.

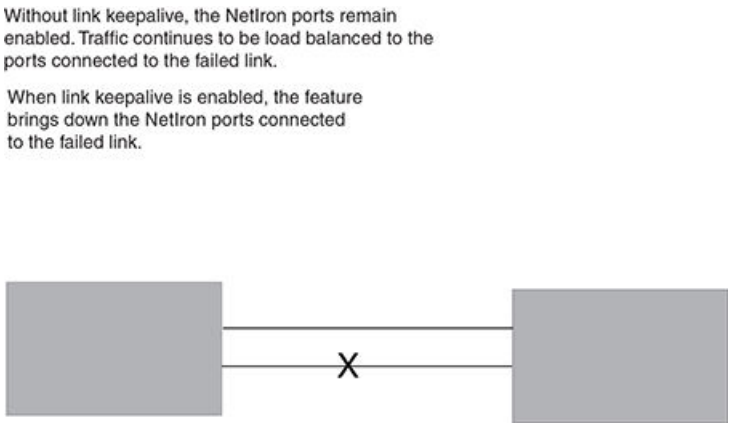
- LP CPU utilization will be higher than normal due to the additional lookup being performed. The following table has the details of CPU usage profiling tested with a simple configuration (plain L3 traffic being sampled without any other protocol/activity like SNMP monitoring, ACL accounting etc in place). This is to provide an estimate of the LP CPU utilization expected when this feature is enabled.

Total Packets hitting the CPU per second	LP CPU utilization
1000	2% to 3%
2000	4% to 5%
4000	8% to 9%
8000	16% to 17%

Configuring Uni-Directional Link Detection

Uni-directional Link Detection (UDLD) monitors a link between two Brocade devices and provides a fast detection of link failures. UDLD brings the ports on both ends of the link down if the link goes down at any point between the two devices. This feature is useful for links that are individual ports and for LAG links. Figure 207 shows an example.

FIGURE 203 UDLD example



By default, ports enabled for UDLD exchange proprietary health-check packets once every 500 ms (the keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for four more intervals. If the port still does not receive a health-check packet after waiting for five intervals, the port concludes that the link has failed and takes the port down.

The Keepalive Interval and Keepalive Retries can be configured to values other than the default, as shown in Changing the keepalive interval on page 848 and Changing the keepalive retries on page 848.

Configuration considerations

This section describes the configuration considerations:

- The feature is supported only on Ethernet ports.
- To configure UDLD on a LAG group, you must configure the feature on each port of the group individually. Configuring UDLD on a LAG group's primary port enables the feature on that port only.
- Dynamic LAG is not supported.
- If you want to configure UDLD on a static LAG group that contains ports on which UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the LAG group, you can again add the UDLD configuration.
- UDLD must be configured on both sides of the link.

Configuring UDLD

You use this command to enable UDLD on a port.

To enable UDLD on a port, enter a command such as the following at the global CONFIG level of the CLI.

```
device(config)# link-keepalive ethernet 1/1
```

To enable the feature on a LAG group, enter commands such as the following.

```
device(config)# link-keepalive ethernet 1/1 ethernet 1/2
device(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.

```
device(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.

```
device(config)# link-keepalive retries 4
```

Syntax: [no] link-keepalive retries num

The *num* parameter specifies the maximum number of times the port will try the health check. You can specify a value from 3 - 10. The default is 5.

UDLD for tagged ports

The default implementation of UDLD sends the packets untagged, even across tagged ports. If the untagged UDLD packet is received by a third-party switch, that switch may reject the packet. As a result, UDLD may be limited only to Brocade devices, since UDLD may not function on third-party switches.

Beginning with Multi-Service IronWare release 04.1.00, you can configure ports to send out UDLD control packets that are tagged with a specific VLAN ID as tagged UDLD control packets. The enhancement also allows third party switches to receive the control packets that are tagged with the specified VLAN.

To allow ports to receive and send UDLD control packets tagged with a specific VLAN ID, enter commands such as the following:

```
device(config)# link-keepalive ethernet 1/18 vlan 22
```

This commands enables UDLD on port 1/18 and allows UDLD control packet tagged with VLAN 22 to be received and sent on port 1/18.

Syntax: [no] link-keepalive ethernet *portnum* [**vlan** *vlan-ID*]

Enter the slot number (if applicable) and the port number of the Ethernet port.Enter the ID of the VLAN that the UDLD control packets can contain to be received and sent on the port. If a VLAN ID is not specified, then UDLD control packets are sent out of the port as untagged packets.

NOTE
You must configure the same VLANs that will be used for UDLD on all devices across the network; otherwise, the UDLD link cannot be maintained.

Displaying UDLD information

Displaying information for all ports

.In the following example, port 1/1 is tagged in VLAN 2 and is configured for tagged UDLD. Port 1/2 is not configured for tagged UDLD:

```
device#show link-keepalive
Total link-keepalive enabled ports: 2
Keepalive Retries: 5    Keepalive Interval: 5 * 100 MilliSec.
Port    Physical Link   Link-keepalive  Logical link    Link-vlan
1/1         down            down                down              2
1/2         down            down                down
device#
```

Syntax: show link-keepalive [**ethernet slot/portnum**]

TABLE 108 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 Brocade port and the directly connected device.
Link-keepalive	Show if the keepalive link is up or down.
Logical Link	The state of the logical link. This is the state of the link between this device port and the device port on the other end of the link. If the states of both Physical Link and Link-keepalive are up, then Logical link is up. If either or both Physical Link and Link-keepalive states are down, then Logical Link displays "down".
Link-vlan	The VLAN that the port is configured to tag the UDLD packets with.

If a port is disabled by UDLD, the change also is indicated in the output of the **show interfaces brief** command. Here is an example.

```
device(config)# show interface brief
Slot/
Port  Link State      Dupl Speed Trunk Tag Priori MAC            Name
1/1   Up   LK-DISABLE
None None  None  No  level0 00e0.52a9.bb00
```

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

Displaying information for a single port

To display detailed UDLD information for a specific port, the **show link-keepalive ethernet** command is valid for a port that is not configured for Tagged UDLD. For a Tagged UDLD port VLAN information is also included. In the following example, port 1/1 is tagged in VLAN 2 and is configured for tagged UDLD. Port 1/2 is not configured for tagged UDLD.

```
device#show link-keepalive ethernet 1/1
Current State   : down          Remote MAC Addr  : 0000.0000.0000
Local Port      : 1/1           Remote Port      : n/a
Local System ID : 1bb3d340      Remote System ID : 00000000
Packets sent    : 0             Packets received : 0
Transitions     : 5             Link-Vlan        :2
device#show link-keepalive ethernet 1/2
Current State   : down          Remote MAC Addr  : 0000.0000.0000
Local Port      : 1/2           Remote Port      : n/a
Local System ID : 1bb3d340      Remote System ID : 00000000
Packets sent    : 0             Packets received : 0
Transitions     : 6
device#
```

TABLE 109 CLI display of detailed UDLD information

This field...	Displays...
Current State	The state of the logical link. This is the link between this device port and the device port on the other end of the link.
Remote MAC Addr	The MAC address of the port or device at the remote end of the logical link.
Local Port	The port number on this router.
Remote Port	The port number on the router at the remote end of the link. NOTE The Remote Port number shown in this parameter reflects the port ID sent by the other router or switch and interpreted by this local router. In cases where this router interprets the port ID different than the router that sent the port ID, the port shown can be incorrect.
Local System ID	A unique value that identifies this router. The ID can be used by technical support for troubleshooting.
Remote System ID	A unique value that identifies the router at the remote end of the link.
Packets sent	The number of UDLD health-check packets sent on this port.
Packets received	The number of UDLD health-check packets received on this port.
Transitions	The number of times the logical link state has changed between up and down.
Link-vlan	The VLAN that the port is configured to tag the UDLD packets with.

The **show interface ethernet***slot/portnum* command also displays the UDLD state for an individual port. In addition, the line protocol state listed in the first line will say "down" if UDLD has brought the port down. Here is an example.

```
device(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.

```
device# 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** command display.

BiDirectional Forwarding Detection (BFD)

BFD provides a rapid forwarding path failure detection service to a Routing Protocol.

BFD provides rapid detection of the failure of a forwarding path by checking that the next-hop device is alive. Without BFD enabled it can take from 3 to 30 seconds to detect that a neighboring device is not operational, causing packet loss due to incorrect routing information at a level unacceptable for real-time applications such as VOIP and video over IP.

Using BFD, you can detect a forwarding path failure in 300 milliseconds or less, depending on your configuration.

A BFD session is automatically established when a neighbor is discovered for a protocol provided that BFD is enabled on the interface on which the neighbor is discovered and BFD is also enabled for the protocol (by interface or globally). Once a session is set-up, each device transmits control messages at a high rate of speed that is negotiated by the devices during the session setup. To provide a detection time of 150 milliseconds, it is necessary to process 20 messages per second of about 70 to 100 bytes each per session. A similar number of messages also need to be transmitted out per session. Once a session is set-up, that same message is continuously transmitted at the negotiated rate and a check is made that the expected control message is received at the agreed frequency from the neighbor. If the agreed upon messages are not received from the neighbor within a short period of time, the neighbor is considered to be down.

For the NetIron CES and NetIron CER device, there are 20 Bidirectional Sessions per LP and 40 Bidirectional sessions system-wide.

NOTE
BFD session establishment on an interface does not start until 90 seconds after the interface comes up. The reason for this delay is to ensure that the link is not effected by unstable link conditions which could cause BFD to flap. This delay time is not user configurable.

The BFD Control Message is an UDP message with destination port 3784.

NOTE
BFD version 0 is not supported in this implementation and BFD version 1 is not compatible with BFD version 0.

NOTE
BFD supports multi-slot LAGs in cases where all BFD packet are transmitted only on a single path which does not change unless the LAG active membership changes. BFD is not be supported on multi-slot LAGs where per-packet switching is used such that the path taken by the BFD packets will vary per packet.

NOTE
When BFD is configured with stringent values of 100/300 msec, BFD may flap when learning a large number of routes.

NOTE
BFD sessions configured with lower timer values may exhibit flaps when configured alongside MACSec on same line card. This issue is a known limitation. However, the BFD sessions are stable with 500ms*3 timer value or more in such scenarios.

Number of BFD sessions supported

The devices have a set limit of 250 BFD sessions per system with a maximum number of 40 sessions per Interface Module. This number is inclusive of the fact that IS-IS and OSPF sessions on an Interface Module will include both Tx and Rx sessions. Consequently, the 40 sessions per Interface Module actually corresponds to 80 sessions where each OSPF and IS-IS session consumes 2 sessions (1 Tx and 1 Rx).

Unlike IS-IS and OSPF however, the Tx and Rx sessions for MPLS BFD can reside on different interface modules. This means that when counting MPLS BFD sessions against the Interface Module maximum, the Tx and Rx sessions must be counted separately. In practice this means that the maximum number of sessions per-Interface Module is 80; where each Tx and Rx session for MPLS BFD is counted as 1 and IS-IS and OSPF BFD sessions are counted as 2 towards a per-Interface Module maximum number of sessions of 80.

> NOTE
> This BFD session support is applicable to XMR series, MLX series, and the MLXe series only.

Configuring BFD parameters

When you configure BFD you must set timing and interval parameters. These are configured on each interface. When two adjacent interfaces with BFD are configured, they negotiate the conditions for determining if the connection between them is still active. The following command is used to set the BFD parameters:

```
device(config-if-e1000-3/1)# bfd interval 100 min-rx 100 multiplier 3
```

Syntax: [no] bfd interval *transmit-time* **min-rx** *receive-time* **multiplier** *number*

The *transmit-time* variable is the interval in milliseconds between which this device will send a BFD message to its peer informing it that it is still operational. This value is specified in milliseconds. Acceptable values are: 50 - 30000.

The *receive-time* variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device waits for this interval for the number of times specified in the *number* variable before determining that the connection to its peer in not operational. Acceptable values are: 50 - 30000.

> NOTE
> The *transit-time* and *receive-time* variables are the intervals desired by the local device. The actual values in use will be the negotiated values.

The *number* variable specifies the number of times in a single sequence that this device will wait to receive a BFD message from its peer before determining that the connection to that peer is not operational. Acceptable values are: 3 - 50.

When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

Disabling BFD Syslog messages

Syslog messages are generated for BFD operations. Logging of these messages is enabled by default. To disable logging of BFD messages use the following command:

```
device(config)# no logging enable bfd
```

Syntax: [no] logging enable bfd

BFD logging is enabled by default. If you disable BFD logging as shown, you can re-enable it by using the **logging enable bfd** command.

Displaying BFD information

You can display BFD information for the device you are logged-in to and for BFD-configured neighbors as described in the following sections.

Displaying BFD information

The following example illustrates the output from the **show bfd** command:

```
device# show bfd
 BFD State: ENABLED Version: 1 Use PBIF Assist: Y
Current Registered Protocols: ospf/0  ospf6/0
All Sessions: Current: 4 Maximum Allowed: 100 Maximum Exceeded Count: 0
LP Sessions: Maximum Allowed on LP: 40  Maximum Exceeded Count for LPs: 0
LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions
1   4/4           2   2/2            3  0/0            4  0/0
5   0/0           6   0/0            7  0/0            8  0/0
9   0/0           10  0/0            11 0/0            12 0/0
13  0/0           14  0/0            15 0/0            16 0/0
BFD Enabled ports count: 2
Port       MinTx       MinRx       Mult Sessions
eth 2/1    100         100         3    2
eth 3/1    100         100         3    2
```

Syntax: show bfd

This display shows the following information.

TABLE 110 Display of BFD information

This field...	Displays...
BFD State	Specifies if BFD is Enabled or Disabled on the device.
Version	Specifies the version of the BFD protocol operating on the device.
Use PBIF Assist	Specifies the status of PCI Bus Interface (PBIF) Assist.
Current Registered Protocols	Specifies which protocols are registered to use BFD on the device. Possible values are mpls/0, ospf/0, ospf6/0, or isis_task/0
All Sessions	
Current:	The number of BFD sessions currently operating on the device.
Maximum Allowed	The maximum number of BFD sessions that are allowed on the device. The maximum number of sessions supported on a device is 250 for Ni-XMR and Ni-MLX and 40 for Ni-CES.
Maximum Exceeded Count	The number of times the request to set up a BFD session was declined because it would have resulted in exceeding the maximum number of BFD sessions allowed on the device.
LP Sessions:	
Maximum Allowed on LP	The maximum number of BFD sessions that are allowed on an interface module. The maximum number of sessions supported on an interface module is 40 for Ni-XMR and Ni-MLX and 20 for Ni-CES
Maximum Exceeded Count for LPs	The number of times the request to set up a BFD session was declined because it would have resulted in exceeding the maximum number of BFD sessions allowed on an interface module.
LP	The number of the interface module that the Current Session Count is displayed for.
Sessions	The number of Transmit (Tx) and Receive (Rx) BFD sessions currently operating on the specified interface module.
BFD Enabled ports count	The number of ports on the device that have been enabled for BFD.

TABLE 110 Display of BFD information (continued)

This field...	Displays...
Port	The port that BFD is enabled on.
MinTx	The interval in milliseconds between which the device desires to send a BFD message from this port to its peer.
MinRx	The interval in milliseconds that this device desires to receive a BFD message from its peer on this port.
Mult	The number of times that the device will wait for the MinRx time on this port before it determines that its peer device is non-operational.
Sessions	The number of BFD sessions originating on this port.

NOTE

On a non POS module, BFD makes use of PBIF transmit assist to send out the packets with the maximum transmit interval of 6.4 seconds.

Displaying BFD neighbor information

The following example illustrates the output from the **show bfd neighbor** command.

```
device# show bfd neighbor
Total number of Neighbor entries: 2
NeighborAddress                       State   Interface Holddown  Interval  R/H
10.14.1.1                             UP      eth 3/1   300000    100000    Y/S
10.2.1.1                              UP      eth 2/1   300000    100000    Y/S
```

Syntax: show bfd neighbor [**interface ethernet slot/port** | **interface ve port-no**]

The **interface ethernet** option displays BFD neighbor information for the specified ethernet interface only.

The **interface ve** option displays BFD neighbor information for the specified virtual interface only.

This display shows the following information.

TABLE 111 Display of BFD information

This field...	Displays...
Total number of Neighbor entries	The number of neighbors that have established BFD sessions with ports on this device.
NeighborAddress	The IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session: Up Down A.DOWN - The administrative down state. INIT - The Init state. UNKNOWN - The current state is unknown.
Interface	The logical port (physical or virtual port) on which the peer is known.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
R/H	R - Heard from Remote. H - singlehop (S) or Multihop (M)

To display BFD Neighbor information in the detailed format use the following command.

```
device# show bfd neighbor detail
Total number of Neighbor entries: 1
NeighborAddress                       State   Interface Holddown  Interval  R/H
10.14.1.1                             UP      ve 50     300000    100000    Y/S
  Registered Protocols(Protocol/VRFID): ospf/0
  Local: Disc: 1, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Remote: Disc: 22, Diag: 7, Demand: 0 Poll: 0
          MinTxInterval: 100000, MinRxInterval: 100000, Multiplier: 3
  Stats: RX: 72089 TX: 72101 SessionUpCount: 1 at SysUpTime: 0:1:30:54.775
  Session Uptime: 0:1:30:6.375, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port: eth 4/1, Vlan Id: 50,Session: Active
Using PBIF Assist: Y
```

Syntax: show bfd neighbor details [*ip-address* | *ipv6-address*]

This display shows the following information.

TABLE 112 Display of BFD neighbor detail information

This field...	Displays...
Total number of Neighbor entries	Total number of BFD sessions.
NeighborAddress	IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session: Up Down A.DOWN - The administrative down state. INIT - The Init state. UNKNOWN - The current state is unknown.
Interface	The logical port on which the peer is known.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
R/H	R - Heard from Remote. H - singlehop (S) or Multihop (M).
Registered Protocols	Specifies which protocols are registered to use BFD on this port.
Local:	
Disc	Value of the "local discriminator" field in the BFD Control Message as used by the local device in the last message sent.
Diag	Value of the "diagnostic" field in the BFD Control Message as used by the local device in the last message sent.
Demand	Value of the "demand" bit in the BFD Control Message as used by the local device in the last message sent.
Poll	Value of the "poll" bit in the BFD Control Message as used by the local device in the last message sent.
MinTxInterval	The interval in microseconds between which the device will send a BFD message from this local neighbor port to its peer.
MinRxInterval	The interval in microseconds that the neighbor device waits to receive a BFD message from its peer on this local port.
Multiplier	The number of times that the neighbor device will wait for the MinRxInterval time on this port before it determines that its peer device is non-operational.

TABLE 112 Display of BFD neighbor detail information (continued)

This field...	Displays...
Remote:	
Disc	Value of the "local discriminator" field in the BFD Control Message as received in the last message sent by the remote peer.
Diag	Value of the "diagnostic" field in the BFD Control Message as received in the last message sent by the remote peer.
Demand	Value of the "demand" bit in the BFD Control Message as received in the last message sent by the remote peer.
Poll	Value of the "poll" bit in the BFD Control Message as received in the last message sent by the remote peer.
MinTxInterval	The interval in milliseconds between which the device will send a BFD message from the remote neighbor port to its peer.
MinRxInterval	The interval in milliseconds that the neighbor device waits to receive a BFD message from its peer on this remote port.
Multiplier	The number of times that the remote neighbor device will wait for the MinRxInterval time on this port before it determines that its peer device is non-operational.
Stats: Rx	Total number of BFD control messages received from the remote peer.
Stats: Tx	Total number of BFD control messages sent to the remote peer.
Stats: SessionUpCount	The number of times the session has transitioned to the UP state.
Stats: SysUpTime	The amount of time that the system has been up.
Session Uptime	The amount of time the session has been in the UP state.
LastSessionDownTimestamp	The system time at which the session last transitioned from the UP state to some other state.
Physical Port	The physical port on which the peer is known.
Vlan Id	The VLAN ID of the VLAN that the physical port is resident on.
Session	
Using PBIF Assist	Y - PBIF Assist is used for this BFD session N- PBIF is not used for this BFD session.

Clearing BFD neighbor sessions

You can clear all BFD neighbor sessions or a specified BFD neighbor session using the following command.

```
device# clear bfd neighbor
```

Syntax: clear bfd neighbor [*ip-address* | *ipv6-address*]

The *ip-address* variable specifies the IPv4 address of a particular neighbor whose session you want to clear BFD.

The *ipv6-address* variable specifies the IPv6 address of a particular neighbor whose session you want to clear BFD.

Executing this command without specifying an IP or IPv6 address clears the sessions of all BFD neighbors.

Configuring BFD for the specified protocol

BFD can be configured for use with the following protocols:

- OSPFv2
- OSPFv3

- IS-IS
- BGP4
- BGP4+

NOTE
BFD is not supported for OSPF v2 or v3 virtual links.

NOTE
BFD brings IS-IS and OSPF down with it when RSTP path-cost changes are made to the switch Alt Discarding port.

Configuring BFD for OSPFv2

You can configure your device for BFD on the OSPFv2 protocol for all OSPFv2 enabled interfaces or for specific interfaces as shown in the following sections.

Enabling BFD for OSPFv2 for all interfaces

You can configure BFD for OSPFv2 on all of a device's OSPFv2 enabled interfaces using the command shown in the following"

```
device# router ospf
device(config-ospf-router)# bfd all-interfaces
```

Syntax: [no] bfd all-interfaces

Although this command configures BFD for OSPFv2 on all of the OSPFv2 enabled interfaces for a device, it is not required if you use the**ip ospf bfd** command to configure specific interfaces. It can be used independently or together with the **ip ospf bfd** command.

Enabling or disabling BFD for OSPFv2 for a specific interface

You can selectively enable or disable BFD on any OSPFv2 interface as shown.

```
device(config-if-e1000-3/1)# ip ospf bfd disable
```

Syntax: ip ospf bfd [**disable**]

The **disable** option disables BFD for OSPF on the interface.

Holdover interval

The BFD holdover interval is supported for single hop sessions. It sets the time by which the BFD session DOWN notification to OSPF is delayed. If within that holdover time, the BFD session is UP then OSPF is not notified of the BFD session flap.

The holdover interval can be configured globally.

If the **bfd holdover-interval** is set to 20 seconds, when a notification is received from BFD that the BFD session has moved to DOWN state, the system waits for 20 seconds before sending the BFD session down notification to OSPFv2 state machine. If the BFD session returns to UP state before the 20 seconds expires, the OSPFv2 state machine is not notified that the BFD session flapped. Otherwise, after 20 seconds the BFD session down notification is passed to the OSPFv2 state machine. If BFD for OSPFv2 is disabled, the request to not use BFD for OSPFv2 is passed to BFD by OSPFv2, BFD acknowledges this request, and the BFD session is removed. If BFD is disabled, OSPFv2 is notified and asks BFD to remove the single hop BFD session on the interface.

NOTE
The benefit of this feature is that OSPF adjacency will not go down if a BFD session is reestablished within the holdover interval preventing disruption to the OSPF routing table.

To configure the BFD down notification delay, enter a command such as the following.

```
device(config-ospf-router)# bfd holdover-interval 10
```

Syntax: [no] bfd holdover-interval *time-seconds*

The *time-seconds* variable is a number between 0 and 20 seconds. The default is 0 seconds.

The **no** option removes the BFD holdover interval from the configuration.

Configuring BFD for OSPFv3

You can configure your device for BFD on the OSPFv3 protocol for all OSPFv3 enabled interfaces or for specific interfaces as shown in the following sections.

Enabling BFD for OSPFv3 for all interfaces

You can configure BFD for OSPFv3 on all OSPFv3 enabled interfaces using the command shown in the following.

```
device(config)# ipv6 router ospf
device(config-ospf6-router)# bfd all-interfaces
```

Syntax: [no] bfd all-interfaces

Although this command configures BFD for OSPFv3 on all of the OSPFv3 enabled interfaces on a device, it is not required if you use the **ipv6 ospf bfd** command to configure specific interfaces. It can be used independently or together with the **ipv6 ospf bfd** command.

Enabling or disabling BFD for OSPFv3 for a specific interface

You can selectively enable or disable BFD on any OSPFv3 interface as shown in the following.

```
device(config-if-e1000-3/1)# ipv6 ospf bfd enable
```

Syntax: ipv6 ospf bfd [**disable** | **enable**]

The **disable** option disables BFD for OSPFv3 on the interface. The **enable** option enables BFD for OSPFv2 on the interface.

Holdover interval

The BFD holdover interval is supported for single hop sessions. It sets the time by which the BFD session DOWN notification to OSPFv3 is delayed. If within that holdover time, the BFD session is UP then OSPFv3 is not notified of the BFD session flap.

The holdover interval can be configured globally.

If the **bfd holdover-interval** is set to 20 seconds, when a notification is received from BFD that the BFD session has moved to DOWN state, the system waits for 20 seconds before sending the BFD session down notification to OSPFv3 state machine. If the BFD session returns to UP state before the 20 seconds expires, the OSPFv3 state machine is not notified that the BFD session flapped. Otherwise, after 20 seconds the BFD session down notification is passed to the OSPFv3 state machine. If BFD for OSPFv3 is disabled, the request to not use BFD for OSPFv3 is passed to BFD by OSPFv3, BFD acknowledges this request, and the BFD session is removed. If BFD is disabled, OSPFv3 is notified and asks BFD to remove the single hop BFD session on the interface.

To configure the BFD down notification delay, enter a command such as the following.

```
device(config-ospf6-router)# bfd holdover-interval 10
```

Syntax: [no] bfd holdover-interval *time-seconds*

The *time-seconds* variable is a number between 0 and 20 seconds. The default is 0 seconds.

The **no** option removes the BFD holdover interval from the configuration.

Configuring BFD for IS-IS

You can configure your device for BFD (for both IPv4 and IPv6 IS-IS neighbors) for the IS-IS protocol for all IS-IS enabled interfaces, or for specific interfaces as shown in the following sections.

> NOTE
> You will not be able to configure a BFD for IS-IS session when one side of the IS-IS adjacency is using IPv4 only and other side is using IPv6 Only.

Enabling BFD for IS-IS for all interfaces

You can configure IS-IS for IS-IS on all S-IS enabled interfaces for a device using this command.

```
device# router isis
device(config-isis-router)# bfd all-interfaces
```

Syntax: [no] bfd all-interfaces

Although this command configures BFD for IS-IS on all IS-IS enabled interfaces on the device, it is not required if you use the**isis bfd** command to configure specific interfaces. It can be used independently or together with the **isis bfd** command.

Enabling or disabling BFD for IS-IS for a specific interface

You can selectively enable or disable BFD on any IS-IS interface as shown in the following.

```
device(config-if-e1000-3/1)# isis bfd
```

Syntax: [no] isis bfd [**disable**]

The **disable** option disables BFD for IS-IS on the interface.

Holdover interval

The BFD holdover interval is supported for single hop sessions. It sets the time by which the BFD session DOWN notification to ISIS is delayed. If within that holdover time, the BFD session is UP then ISIS is not notified of the BFD session flap.

The holdover interval can be configured globally.

If the **bfd holdover-interval** is set to 20 seconds, when a notification is received from BFD that the BFD session has moved to DOWN state, the system waits for 20 seconds before sending the BFD session down notification to ISIS state machine. If the BFD session returns to UP state before the 20 seconds expires, the ISIS state machine is not notified that the BFD session flapped. Otherwise, after 20 seconds the BFD session down notification is passed to the ISIS state machine. If BFD for ISIS is disabled, the request to not use BFD for ISIS is passed to BFD by ISIS, BFD acknowledges this request, and the BFD session is removed. If BFD is disabled, ISIS is notified and asks BFD to remove the single hop BFD session on the interface.

To configure the BFD down notification delay, enter a command such as the following.

```
device(config-isis-router)# bfd holdover-interval 10
```

Syntax: [no] bfd holdover-interval *time-seconds*

The *time-seconds* variable is a number between 0 and 20 seconds. The default is 0 seconds.

The **no** option removes the BFD holdover interval from the configuration.

Configuring BFD for BGP4

You can configure your device for BFD for BGP4. BGP4 supports IPv4 and IPv6 IBGP and EBGP peers. These peers can be directly connected or multihop. BFD for BGP4 supports single hop and multihop BFD on Ethernet, POS and Virtual Interfaces. BFD for BGP4 is not supported on loopback, tunnel (including IGP shortcut) and management interfaces.

BFD for BGP4 global configuration - Using this configuration, you can enable and disable BFD BGP4 on a global level. In addition, you can use the global command to set revised default values for the transmit interval, receive interval, and for the detection time multiplier for all BFD multihop BGP4 sessions.

BFD for BGP4 at global, peer, and peer group configuration - Using this configuration, you can enable and disable BFD for individual peers or peer groups. You can also change the values for the transmit interval, receive interval, and for the detection time multiplier. If these values are not specified at this level, they are obtained from the values configured at the global level.

BFD for BGP4, which is disabled by default, can be enabled or disabled at the global BGP router level or for each individual peer or peer group. The hierarchy for BFD for BGP4 is as follows:

- Peer and peer group parameters can be configured but will not take effect until BFD for BGP4 has been enabled.
- Peer configurations will override global and peer group configurations.
- Peer group configurations will override global configurations
- The **bfd-enable** command under **router bgp** overrides all other BGP4 BFD configurations

Enabling BFD for BGP4 globally

By default, BFD for BGP4 is disabled and can be first enabled globally and then on each peer. To enable BFD for BGP4 globally, enter commands such as the following.

```
device# router bgp
device(config-bgp)# bfd-enable
```

To disable BFD for BGP globally and terminate all BFD sessions used by BGP4, enter commands such as the following

```
device# router bgp
device(config-bgp)# no bfd-enable
```

Syntax: [no] bfd-enable

> NOTE
> If BFD for BGP4 is globally disabled and then enabled, the original BFD sessions for BGP4 may not be available, depending on whether or not the maximum BFD sessions limit has been reached. When a BFD session for BGP4 is disabled, the session will be removed but BGP4 peering will not go down. The remote BFD peer will be informed that BFD use is disabled.

Setting the transmit, receive, and detection time multiplier at the global level

When using BFD for BGP4, you must configure BFD globally at the **router BGP** level. You can also use this configuration to set new default values for the transmit interval, receive interval, and for the detection time multiplier.

For a single hop EBGP session, the BFD parameters configured under interface will be used because the BFD session for single hop is also shared with other applications. To create a BFD session for a single hop BGP4 session, you must have BFD enabled and the timers configured for the interface on which single hop BGP4 peering is established.

> NOTE
> For multihop BFD sessions, BFD does not have to be enabled for any of the interfaces, and the BFD timers need not be configured, since the default values can be used.

The timers parameters **min-tx**, **min-rx** and **multiplier** can also be configured for each peer and peer group and will override the global configuration.

To configure a multi hop EBGP or IBGP session, enter a command such as the following.

```
device(config)# router bgp
device(config-bgp)# bfd-enable
device(config-bgp)# bfd min-tx 500 min-rx 500 multiplier 5
```

Syntax: [no] bfd min-tx *transmit-time* **min-rx** *receive-time* **multiplier** *number*

The *transmit-time* variable is the interval in milliseconds between which this device will send a BFD message to its peer informing it that it is still operational. Acceptable values are: 50 - 30000. Default value: 1000 (unless changed at the global level

The *receive-time* variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device will wait for this interval for the number of times specified in the *number* variable before determining that the connection to its peer in not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level)

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device will wait to receive a BFD message from its peer before determining that the connection to that peer is not operational. This value is set at the *number* variable. Acceptable values are 3 - 50. The default value is 3.

When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

The **no** option globally removes the BFD for BGP4 configuration from the device.

Holdover interval

The BFD holdover interval is supported for both single hop and multihop sessions. It sets the time by which the BFD session DOWN notification to BGP4 is delayed. If within that holdover time, the BFD session is UP then BGP4 is not notified of the BFD session flap.

The holdover interval can be configured globally, on each peer, or peer-group.

If the **bfd holdover-interval** is set to 20 seconds, when a notification is received from BFD that the BFD session has moved to DOWN state, the system waits for 20 seconds before sending the BFD session down notification to BGP4 state machine. If the BFD session returns to UP state before the 20 seconds expires, the BGP4 state machine is not notified that the BFD session flapped. Otherwise, after 20 seconds the BFD session down notification is passed to the BGP4 state machine. If BFD for BGP4 is disabled, the request to not use BFD for BGP4 is passed to BFD by BGP4, BFD acknowledges this request, and the BFD session is removed. If BFD is disabled, BGP4 is notified and asks BFD to remove the single hop BFD session on the interface.

To configure the BFD down notification delay, enter a command such as the following.

```
device(config-bgp)# bfd holdover-interval 20
```

Syntax: [no] bfd holdover-interval *time-seconds*

The *time-seconds* variable is a number between 0 and 30 seconds. The default is 0 seconds.

The **no** option removes the BFD for BGP4 holdover interval from the configuration.

Enabling BFD for a BGP4 peer group

To enable BFD and create a peer group for BGP4, you must first create the peer group, then enable BFD for the peer group by entering commands such as the following.

```
device(config-bgp)# neighbor pg1 peer-group
device(config-bgp)neighbor pg1 fail-over bfd-enable
```

Syntax: [no] neighbor *name* **peer group**

Syntax: [no] neighbor *name* **fail-over bfd-enable**

The *name* variable specifies peer-group name of a particular neighbor.

The **no** option removes the BFD for BGP4 peer group from the configuration.

Enabling BFD timers for a BGP4 peer group

To enable BFD timers for a BGP4 peer group, you must first create the peer group, then enable BFD timers for the peer group by entering commands such as the following.

```
device(config-bgp)# neighbor pg1 peer-group
device(config-bgp)# neighbor pg1 bfd min-tx 500 min-rx 500 multiplier 5
```

Syntax: [no] neighbor *name* **peer group**

Syntax: [no] neighbor *name* **bfd min-tx** *transmit-time* **min-rx** *receive-time* **multiplier** *number*

The *name* variable specifies peer-group name of a particular neighbor.

The *transmit-time* variable is the interval in milliseconds during which this device will send a BFD message to its peer informing it that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The *receive-time* variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device waits for this interval for the number of times specified in the *number* variable before determining that the connection to its peer in not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that the device waits to receive a BFD message from its peer before determining that the connection to that peer is not operational. This value is set at the *number* variable. Acceptable values are 3 - 50. The default value is 3.

When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

The **no** option removes the BFD timers for the peer group from the configuration.

Enabling BFD for a specific BGP4 peer

To enable BFD for BGP4 for a specific neighbor or peer, enter a command such as the following

```
device(config-bgp)# neighbor 10.14.1.1 fail-over bfd-enable
```

Syntax: [no] neighbor *ipv4-address* | *ipv6-address* **fail-over bfd-enable**

The *ipv4-address* and *ipv6-address* variables specify the IPv4 or IPv6 address of a particular neighbor or peer.

The **no** option removes the BFD for BGP4 peer from the configuration.

Disabling BFD for a specific BGP4 peer

To disable BFD for BGP4 for a specific peer, enter a command such as the following.

```
device(config-bgp)# neighbor 10.14.1.1 fail-over bfd-disable
```

Syntax: [no] neighbor *ipv4-address* | *ipv6-address* **fail-over bfd-disable**

The *ipv4-address* and *ipv6-address* variables specify the IPv4 or IPv6 address of a particular neighbor or peer.

The **no** option removes the BFD specific peer from the configuration.

Enabling BFD timers for a specific BGP4 peer

To enable BFD timers for a specific neighbor or peer for BGP4, you must first configure the bfd timers, and set the holdover interval by entering commands such as the following.

```
device(config-bgp)# neighbor 10.14.1.1 bfd min-tx 500 min-rx 500 multiplier 5
device(config-bgp)# bfd holdover-interval 20
```

Syntax: [no] neighbor *ipv4-address* | *ipv6-address* **bfd min-tx** *transmit-time* **min-rx** *receive-time* **multiplier**

Syntax: [no] bfd holdover-interval *time-seconds*

The *ipv4-address* and *ipv6-address* variables specify the IPv4 or IPv6 address of a particular neighbor or peer.

The *time-seconds* variable is a number between 0 and 30 seconds. The default is 0 seconds.

The *transmit-time* variable is the interval in milliseconds between which this device will send a BFD message to its peer informing it that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The *receive-time* variable is the interval in milliseconds that this device waits to receive a BFD message from its peer. The device will wait for this interval for the number of times specified in the *number* variable before determining that the connection to its peer is not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device will wait to receive a BFD message from its peer before determining that the connection to that peer is not operational. This value is set at the *number* variable. Acceptable values are 3 - 50. The default value is 3.

When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

The **no** option removes the BFD for BGP configuration for the peer.

Displaying BFD for BGP4

You can display BFD for BGP4 information for the device you are logged in to and for BFD configured neighbors as described in the following sections.

Displaying BFD information

The following example illustrates the output from the **show bfd** command:

```
device# show bfd
```

BFD State: ENABLED Version: 1 Use PBIF Assist: Y

Current Registered Protocols: **bgp/1** ospf6/0 ospf/0 **bgp/0**

All Sessions: Current: 0 Maximum Allowed: 100 Maximum Exceeded Count: 0

LP Sessions: Maximum Allowed on LP: 20 Maximum Exceeded Count for LPs: 0

LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions LP Tx/Rx Sessions

1 0/0 2 0/0 3 0/0 4 0/0

5 0/0 6 0/0 7 0/0 8 0/0

9 0/0 10 0/0 11 0/0 12 0/0

13 0/0 14 0/0 15 0/0 16 0/0

BFD Enabled ports count: 0

Syntax: show bfd

This display shows the following information.

TABLE 113 Display of BFD information

This field...	Displays...
BFD State	Specifies if BFD is Enabled or Disabled on the device.
Version	Specifies the version of the BFD protocol operating on the device.
Using PBIF Assist	Specifies the status of the PCI Bus Interface (PBIF) Assist.
Current Registered Protocols	Specifies which protocols are registered to use BFD on the device. Possible values are mpls/0, ospf/0, ospf6/0, or isis/0 or bgp/0
All Sessions	
Current:	The number of BFD sessions currently operating on the device.
Maximum Allowed	The maximum number of BFD sessions that are allowed on the device. The maximum number of sessions supported on a device is 250 for Ni-XMR and Ni-MLX and 40 for Ni-CES.
Maximum Exceeded Count	The number of times the request to set up a BFD session was declined because it would have exceeded the maximum number of BFD sessions allowed on the device.
LP Sessions:	
Maximum Allowed on LP	The maximum number of BFD sessions allowed on an interface module. The maximum number of sessions supported is 40 for Ni-XMR and Ni-MLX, and 20 for Ni-CES
Maximum Exceeded Count for LPs	The number of times the request to set up a BFD session was declined because it would have exceeded the maximum number of BFD sessions allowed on an interface module.
LP	The number of the interface modules for which the Current Session Count is displayed.
Sessions	The number of Transmit (Tx) and Receive (Rx) BFD sessions currently operating on the specified interface module.
BFD Enabled ports count	The number of ports that have been enabled for BFD.
Port	The port on which BFD is enabled.
MinTx	The interval in milliseconds during which the device sends a BFD message from this port to its peer.
MinRx	The interval in milliseconds during which this device can receive a BFD message from its peer on this port.
Mult	The number of times the device will wait for the MinRx time on this port before it determines that the peer device is non-operational.
Sessions	The number of BFD sessions originating on this port.

Displaying BFD applications

The following example illustrates the output from the **show bfd applications** command.

```
device# show bfd applications
```

Registered Protocols Count: 3

Protocol VRFID Parameter HoldoverInterval

isis 0 0 2

ospf6 0 1 10

ospf 0 0 5

TABLE 114 Display of BFD applications information

This field...	Displays...
Protocol	Which protocols are registered to use BFD on the device.
VRFID	The VRFID of the protocol.
Parameter	The parameter value passed by the protocol during registration with BFD.
Holdover Interval	The time by which the BFD session DOWN notification is delayed. If within that holdover time, the BFD session is UP then it is not notified of the BFD session flap.

Displaying BFD for BGP neighbor information

The following example illustrates the output from the **show bfd neighbor bgp detail** command for the MLX series and XMR series devices.

device# show bfd neighbor bgp detail

```
Total Entries:4 R:RxRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress                        State  Interface Holddown   Interval R/H
10.101.101.100                        UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 26, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 7, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 14682 TX: 12364 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:50.600, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NeighborAddress                        State  Interface Holddown   Interval R/H
10.100.100.100                        UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 27, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 8, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 14232 TX: 12046 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:49.650, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NeighborAddress                        State  Interface Holddown   Interval R/H
10.1.1.1                               UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 28, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 9, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 15652 TX: 12044 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:48.725, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
NeighborAddress                        State  Interface Holddown   Interval R/H
10.102.102.100                        UP     ve 3      3000000    1000000  Y/M
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 29, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Remote: Disc: 10, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 14232 TX: 12044 SessionUpCount: 1 at SysUpTime: 0:2:46:24.725
  Session Uptime: 0:1:37:48.550, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/1,RX: eth 1/1,Vlan Id: 3
device#
```

The following example illustrates the output from the **show bfd neighbor bfd detail** command for the CES series and CER series devices.

```
device#show bfd neighbor detail
Total Entries:1 R:RXRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
```

```
NeighborAddress                         State  Interface Holddown   Interval R/H
fe80::224:38ff:fe79:9310                UP     eth 1/17  1500000    500000   Y/S
  Registered Protocols(Protocol/VRFID): bgp/0
  Local: Disc: 8, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 500000, MinRxInterval: 500000, Multiplier: 3
Remote: Disc: 2, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 500000, MinRxInterval: 500000, Multiplier: 3
Stats: RX: 160394 TX: 142648 SessionUpCount: 1 at SysUpTime: 5:17:14:13.225
  Session Uptime: 0:17:49:42.100, LastSessionDownTimestamp: 0:0:0:0.0
  Physical Port:TX: eth 1/17,RX: eth 1/17,Vlan Id: 1
  Using PBIF Assist: Y
device#
```

Syntax: show bfd neighbor bgp [*ipv4-address* | *ipv6-address* | **detail**]

The *ipv4-address* and *ipv6-address* options display BFD neighbor information for the BGP specified IPv4 or IPv6 neighbor only.

The **detail** option displays BFD neighbor information for all BGP neighbors.

This display contains the following information.

TABLE 115 Display of BFD neighbor detail information

This field...	Displays...
Total number of Neighbor entries	Total number of BFD sessions.
NeighborAddress	IPv4 or IPv6 address of the remote peer.
State	The current state of the BFD session: Up Down A.DOWN - The administrative down state. INIT - The Init state. UNKNOWN - The current state is unknown.
Interface	The logical port on which the peer is known.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
R/H	Heard from remote, values: Y or N where Y stand for Yes and N stand for No; H- Single hop/Multihop Values are S and M where S stand for Single Hop and M stand for MultiHop.
Registered Protocols	Specifies which protocols are registered to use BFD on this port.
Local:	
Disc	Value of the local discriminator field in the BFD Control Message as used by the local device in the last message sent.
Diag	Value of the diagnostic field in the BFD Control Message as used by the local device in the last message sent.
Demand	Value of the demand bit in the BFD Control Message as used by the local device in the last message sent.
Poll	Value of the poll bit in the BFD Control Message as used by the local device in the last message sent.
MinTxInterval	The interval in microseconds during which the device will send a BFD message from this local neighbor port to the peer.
MinRxInterval	The interval in microseconds that the neighbor device waits to receive a BFD message from the peer on this local port.

TABLE 115 Display of BFD neighbor detail information (continued)

This field...	Displays...
Multiplier	The number of times the neighbor device will wait for the MinRxInterval time on this port before it determines the peer device is non-operational.
Remote:	
Disc	Value of the local discriminator field in the BFD Control Message as received in the last message sent by the remote peer.
Diag	Value of the diagnostic field in the BFD Control Message as received in the last message sent by the remote peer.
Demand	Value of the demand bit in the BFD Control Message as received in the last message sent by the remote peer.
Poll	Value of the poll bit in the BFD Control Message as received in the last message sent by the remote peer.
MinTxInterval	The interval in milliseconds during which the device will send a BFD message from the remote neighbor port to the peer.
MinRxInterval	The interval in milliseconds that the neighbor device waits to receive a BFD message from the peer on this remote port.
Multiplier	The number of times that the remote neighbor device will wait for the MinRxInterval time on this port before it determines that the peer device is non-operational.
Stats: Rx	Total number of BFD control messages received from the remote peer.
Stats: Tx	Total number of BFD control messages sent to the remote peer.
Stats: SessionUpCount	The number of times the session has transitioned to the UP state.
Stats: SysUpTime	The amount of time that the system has been up.
Session Uptime	The amount of time the session has been in the UP state.
LastSessionDownTimestamp	The system time at which the session last transitioned from the UP state to some other state.
Physical Port	The physical port on which the peer is known.
Vlan Id	The VLAN ID of the VLAN on which the physical port is resident.

Displaying summary neighbor information

Support for BFD for BGP neighbors is highlighted in the bold text in the following output for the **show ip bgp neighbors** command.

Neighbor AS4 Capability Negotiation:

As-path attribute count: 2

Outbound Policy Group:

ID: 1, Use Count: 3

BFD:Enabled,BFDSessionState:UP,Multihop:Yes

LastBGP-BFDEvent:RX:Up,BGP-BFDError:No Error

NegotiatedTime(msec):Tx:1000000,Rx:1000000,BFDHoldTime:3000000

HoldOverTime(sec) Configured:22,Current:0,DownCount:0

TCP Connection state: ESTABLISHED, flags:00000044 (0,0)

Maximum segment size: 1460

BFD for RSVP-TE LSP

BFD provides a mechanism to detect data plane failure for MPLS LSP in the order of sub-second. Unlike MPLS LSP ping where MPLS control plane can be verified against the data plane, BFD is used only to detect data plane failure. There are a couple advantages in using BFD instead of LSP ping for MPLS data plane failure detection. BFD provides a faster failure detection mechanism because it does not require control plane verification. BFD can also be used to detect faults for a large number of LSPs without manual trigger for each LSP the way LSP ping does.

BFD session for RSVP LSP is very similar to the BFD session set up for ISIS and OSPF with the following differences:

- MPLS BFD session is bootstrapped using LSP ping.
- The IP TTL for transmitted MPLS BFD control packets from ingress LSR to egress LSR must be set to 1 instead of 255.
- After MPLS BFD session is up, the local discriminator and the source IP address are not allowed to change without bringing down the MPLS BFD session.
- The transmit and receive portion of the session can be on different LPs because the LSP is unidirectional and the returned path from egress to ingress LSR depends on IP routing.

MPLS BFD is set up between the ingress and egress LSR. The MPLS BFD session uses an asynchronous operating mode without echo function. BFD demands an operating mode and echo function that are not currently defined for MPLS BFD. Authentication is not supported (it is currently not supported for OSPF or ISIS either).

BFD configuration on the MLX/XMR products are provided at the global MPLS configuration level and at the LSP level as follows:

- The global configuration provides you a convenient way to enable and disable BFD for all LSPs that have BFD enabled (this is similar to how it is provided for ISIS and OSPF). In addition, the you can change the default settings for transmit and receive intervals and detection time multiplier to be used for all BFD sessions. On egress LSR, this global setting is used to decide whether a BFD session starts upon receiving MPLS echo request with BFD Discriminator TLV and the time intervals to be included in the BFD control packet are to be sent back to the ingress LSR.
- Under the LSP configuration, you are able to enable or disable BFD and change the default settings for minimum transmit and receive intervals as well as detection time multiple. If those parameters are not specified, the values from the global configuration are used.

BFD can be enabled or disabled without program exit at the global MPLS level or for each individual LSP without affecting the LSP operational status. In addition, the BFD parameters can also be changed without program exit This does not change the state of the BFD session.

One BFD session is created for each non-redundant LSP. BFD session is associated with the active path which can be normal or protected, or detour path. For redundant LSP, a separate BFD session is created for the currently active secondary path.

BFD session creation

On ingress, one BFD session is created for each LSP. A BFD session is created after the LSP comes UP. The BFD session status is displayed as part of the show LSP output. When the BFD session is not UP, a failure reason is displayed as part of the show LSP detail output. Possible failure reasons include exceeding the maximum number of BFD sessions the system can support or the global BFD configuration is disabled. In the case where BFD session is not brought up because BFD packet from the egress LSR is not received, the MPLS Echo Request with the BFD Discriminator TLV is resent until the session is UP. The retry timer is exponentially backed off.

On egress, a BFD session is created after an MPLS Echo Request is received with the BFD Discriminator TLV and the MPLS BFD enabled globally and the maximum number of BFD sessions has not been reached yet. The BFD session created on egress LSR also is counted towards the maximum number of BFD sessions. When the number of BFD sessions has reached a maximum, neither the MPLS Echo Reply nor the BFD control packet are sent. The ingress LSR will retry.

Because the source IP address cannot be changed for MPLS BFD session after the session has comes up, the LSR-ID is used as the source IP address for all MPLS BFD packets. This guarantees that the session does not go down when the LSP path switch occurs.

FRR LSP

Only one BFD session is created for a FRR LSP. When a switchover from protected to detour path occurs, and when the detour is on a different LP, the BFD session is moved to the LP where the detour path resides. The BFD session can go down when the LP already has a maximum number of BFD sessions running. When the detour is on the same LP, the outgoing interface and label stack is updated on the existing LP.

A BFD session is not created for detour path originating on a transit LSR.

Redundant LSP

One BFD session is created for a primary path of a redundant LSP. When the secondary path is hot-standby, a separate BFD session is created for it if and only if BFD is enabled for a secondary path. The two sessions operate independently.

Adaptive LSP

When a new instance of an adaptive LSP comes UP, a BFD session automatically moves to a new LP when the new instance is created on a different LP. Otherwise, the local outgoing interface and label stack updates on the existing LP. When the BFD session needs to be moved to a different LP, it is possible that the BFD session may be down when the LP already has maximum number of BFD sessions running.

BFD session deletion

A BFD session is deleted when any of the following events happen:

- LSP goes down
- BFD is disabled for the LSP
- BFD is disabled for all LSP (example: through global MPLS configuration)

BFD session modification

You can change the BFD parameters globally or for an individual LSP without program shutdown without affecting the operational status of the LSP and the BFD session. When you change the global configuration, the change is applied to all egress MPLS BFD sessions and only to those ingress BFD sessions whose parameters are derived from the global configuration.

BFD session down handling

When a BFD session for a LSP goes down on ingress LSR because BFD detection time has expired, it may trigger path switchover if possible (protected to detour or primary to secondary path switchover). In the case where there is no alternative path, the LSP is brought down and the BFD session is deleted. The LSP goes through the normal retry mechanism in order to come back UP.

On egress LSR, BFD session down does not have any impact on the RSVP session.

Configuring BFD for RSVP-TE LSPs

BFD can be configured for use with RSVP-LSPs to detect data plane failures for MPLS LSPs. Although the LSP ping facility can also be used for this purpose, BFD provides the following advantages:

- BFD provides faster failure detection because it does not required control plane verification, which is required by LSP ping.
- BFD can be used to detect faults on a large number of LSPs without requiring manual interaction, which is required by LSP ping.

BFD configuration for RSVP-TE LSPs is performed at the global and LSP levels, as described:

- **BFD for RSVP-TE LSPs global configuration** - allows you to enable and disable BFD on all of the RSVP-TE LSPs that have been configured for BFD at the LSP level. In addition, use the global command to set revised default values for the transmit interval, receive interval, and for the detection time multiplier for all BFD sessions on RSVP-TEs. This configuration command can also be used as a convenient method to turn BFD for MPLS on or off.
- **BFD for RSVP-TE LSPs configuration at the LSP level** - allows you to enable and disable BFD for individual RSVP-TE LSPs. You can also change the values for the transmit interval, receive interval, and for the detection time multiplier from the default values. If these values are not specified at this level, they are obtained from the values configured at the global level.

BFD, which is disabled by default, can be enabled or disabled at the global MPLS level, or for each individual LSP without affecting the LSP operational status. BFD parameters can also be changed without changing the state of the BFD session.

BFD for RSVP-TE LSPs operates with Fast ReRoute (FRR), Redundant, and Adaptive LSPs as described:

- **FRR LSPs –** Only one BFD session is created for an FRR LSP. A separate BFD session is not created for the detour path. When a switchover from a protected to a detour path occurs, the detour path resides on another interface module, and the BFD session is moved to that interface module. The BFD session can go down if the interface module has already reached the maximum number of BFD sessions. If the detour path is on the same interface module, the outgoing interface and label stack are updated on that interface module. A BFD session is not created for a detour path originated on a transit LSR.
- **Redundant LSPs –** One BFD session is created for the primary path of a redundant LSP. If the secondary path is in the hot-standby condition, a separate BFD session is created for it, but only if BFD is enabled on the secondary path. The two sessions operate independently.
- **Adaptive LSPs –** If a new instance of an adaptive LSP comes up on a different interface module, its BFD session is automatically created on that module. Otherwise, the local outgoing interface and label stack are updated on the existing interface module. When a BFD session is moved to a different interface module, the BFD session may be brought down if the interface module has already reached the maximum number of BFD sessions allowed on it.

BFD session support per-router and per-interface module

There is a limit to the number of BFD sessions available on a per-router and per-interface module basis as described:

- **per-router** - A maximum number of 250 BFD sessions are permitted per device
- **per-interface module** - A maximum number of 80 BFD sessions (Tx or Rx) are permitted per-interface module

These limitations are inclusive of any BFD sessions created for OSPFv2 or v3 and IS-IS. If creating a BFD session will exceed these limits, the session will be denied. For a detailed description of how to calculate the number of BFD sessions supported, refer to Number of BFD sessions supported on page 854.

BFD session creation

On ingress, one BFD session is created for each LSP after the LSP comes up. The BFD session status is then displayed in the output of the **show lsp** command. If the BFD session is not up, a failure reason is displayed in the output of the **show lsp** command. Possible reasons why a BFD session may fail to come up include exceeding the maximum supported number of BFD sessions, or if the global BFD configuration is disabled. If a BFD session does not come up because a BFD packet from the egress LSR is not received, an MPLS Echo Request with a BFD Discriminator TLV is resent until the session does come up. The retry timer is exponentially backed off.

On egress, a BFD session is created after the following sequence of events.

1. An MPLS Echo Request is received with a BFD Discriminator TLV
2. MPLS BFD is enabled globally
3. The maximum number of BFD sessions available on the device has not been reached.

NOTE
A BFD session created on an egress LSR is counted toward the maximum supported number of BFD sessions.

If the number of BFD sessions has reached the su pop rt ed maximum for the device, no MPLS Echo Reply or BFD control packet is sent. The ingress LSR will retry.

Because the source IP address cannot be changed for an MPLS BFD session after the session has come up, the LSR-ID is used as the source IP address for all MPLS BFD packets. This ensures that the session will not go down when an LSP path switch occurs.

BFD session down behavior

When a BFD session for an LSP goes down on an ingress LSR because the BFD detection time has expired, one of the following path switchovers will be triggered; from the protected path to the detour path, or from the primary path to the secondary path. In configurations with no alternative path, the LSP is brought down and the BFD session is deleted. The LSP then follows the normal retry procedures to come back up. On an egress LSR, a down BFD session does not have any impact on the RSVP session.

AdminDown State

The AdminDown mechanism in BFD is intended to signal that the BFD session is being taken down for administrative purposes, and the session state is not indicative of the activity of the data path. Therefore, a system should not indicate a connectivity failure to a client if either the local session state or the remote session state (if known) transitions to AdminDown when that client has an independent means of activity detection (typically, control protocols).

If a client does not have any independent means of activity detection, a system should indicate a connectivity failure to a client, and assume the semantics of Down state, if either the local or remote session state transitions to AdminDown. Otherwise, the client will not be able to determine whether the path is viable, if not unfortunate results may occur.

Reaction to BFD Session State Changes

If a BFD session transitions from Up state to AdminDown, or the session transitions from Up to Down because the remote system is indicating that the session is in state AdminDown, clients should not take any control protocol action.

BFD session deletion

A BFD session is deleted when any of the following events occur:

- An LSP goes down
- BFD is disabled for the LSP
- BFD is disabled for all LSPs (using the global configuration)

These events are described in the following sections.

Enabling BFD for RSVP-TE LSPs at the global level

When using BFD for RSVP-TE LSPs, you must configure BFD globally at the **router mpls** level. You can also use this configuration to set new default values for the transmit interval, receive interval, and for the detection time multiplier, as shown.

```
device(config)# router mpls
device(config-mpls)# bfd
device(config-mpls)# min-tx 500 min-rx 500 multiplier 5
```

Syntax: [no] min-tx *transmit-time* **min-rx** *receive-time* **multiplier** *number*

The *transmit-time* variable is the interval in milliseconds during which this device sends a BFD message to the peer informing it that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The *receive-time* variable is the interval in milliseconds the device waits to receive a BFD message from the peer. The device waits for the number of times specified in the *number* variable before determining that the connection to the peer in not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device will wait to receive a BFD message from the peer before determining that the connection to that peer is not operational. This value is set at the *number* variable. Acceptable values are 3 - 50. The default value is 3.

When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

The **no** option globally removes the BFD for RSVP-TE LSPs configuration from the device.

NOTE
BFD parameters configured globally can be changed dynamically without affecting the operational status of the LSP and the BFD session. When you make changes to the global configuration, the changes are applied to all egress MPLS BFD sessions. and only to the ingress BFD sessions with parameters that are derived from the global configuration.

Enabling BFD for a specific RSVP-TE LSP

When you configure BFD globally, you must also configure it locally for the individual LSPs on which you want it to operate. You can also set separate values for the transmit interval, receive interval, and for the detection time multiplier for the specified LSP. The following example enables BFD for the LSP named blue and sets new parameter values.

```
device(config)# router mpls
device(config-mpls)# lsp blue
device(config-mpls-lsp-blue)# bfd
device(config-mpls-lsp-blue-bfd)# min-tx 500 min-rx 500 multiplier 5
```

Syntax: [no] min-tx *transmit-time* **min-rx** *receive-time* **multiplier** *number*

The *transmit-time* variable is the interval in milliseconds during which this device sends a BFD message to the peer announcing that it is still operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The *receive-time* variable is the interval in milliseconds this device waits to receive a BFD message from the peer. The device waits for the number of times specified in the *number* variable before determining that the connection to the peer is not operational. Acceptable values are 50 - 30000. The default value is 1000 (unless changed at the global level).

The **multiplier** option allows you to specify a value for the number of times in a single sequence that this device waits to receive a BFD message from the peer before determining that the connection to that peer is not operational. This value is set at the *number* variable. Acceptable values are 3 - 50. The default value is 3.

When Brocade NetIron CER Series or Brocade NetIron CES Series devices are heavily loaded or under stress, BFD sessions may flap if the configured BFD interval is less than 500 milliseconds with a multiplier value of 3.

The **no** option globally removes the BFD for RSVP-TE LSPs configuration from the device.

NOTE
BFD parameters configured for a specific LSP can be changed dynamically without affecting the operational status of the LSP and the BFD session.

Using this command you can also configure BFD for the secondary path of an LSP as shown in the following example.

```
device(config)# router mpls
device(config-mpls)# lsp blue
```

```
device(config-mpls-lsp-blue)# secondary-path alt_sf_to_sj
device(config-mpls-lsp-blue-sec-path)# bfd
```

Enabling the IP router alert option

NOTE
The **set-router-alert-option** command is supported only for NetIron XMR and NetIron MLX devices.

The **set-router-alert-option** command sets the IP router alert option for MPLS BFD packets sent from the ingress device to the egress device. Brocade devices support the IP router alert option as defined in RFC 2113. To enable router alert option, enter the following command under the LSP BFD configuration.

```
device(config)# router mpls
device(config-mpls)# lsp blue
device(config-mpls-lsp-blue)# bfd
device(config-mpls-lsp-blue-bfd)#set-router-alert-option
```

Syntax: [no] set-router-alert-option

By default, the router alert option is disabled.

The router alert option configuration is only displayed when BFD is enabled for LSP. This example shows the router option enabled for LSP blue.

```
device(config-mpls-lsp-blue-bfd)#show mpls lsp name blue
LSP blue, to 0.0.0.0
  From: (n/a), admin: DOWN, status: DOWN
  Times primary LSP goes up since enabled: 0
  Metric: 0, number of installed aliases: 0
  Maximum retries: 0, no. of retries: 0
  Pri. path: NONE, up: no, active: no
  Setup priority: 7, hold priority: 0
  Max rate: 0 kbps, mean rate: 0 kbps, max burst: 0 bytes
  Constraint-based routing enabled: yes
  Tie breaking: random, hop limit: 0
  Active Path attributes:
   BFD session status: DOWN(LSP down)
    Config params: global, min-tx: 50, min-rx: 50, multiplier: 5
    Set router alert option: yes
```

Configuring time delay for setup of BFD single-hop session

You can define a time delay for establishing the BFD single hop session after the port is enabled.

Using the command you can delay the setup of BFD single hop session.

```
Brocade(config)#bfd sh-session-setup-delay 40
```

Syntax: [no] bfd sh-session-setup-delay *seconds*

By default, the time delay to establish the single hop session is set to 180 seconds. The **no** form of the command removes the time delay for the session.

The *seconds* value is the time delay in seconds. You can specify a value between 0 and 600 seconds. The default value is 90 seconds.

Configuring time delay for setup of BFD multihop session

You can define a time delay for establishing the BFD multihop session after the system initializes.

Using the command you can delay the setup of BFD multihop session.

```
Brocade(config)#bfd mh-session-setup-delay 90
```

Syntax: [no] bfd mh-session-setup-delay*seconds*

By default, the time delay to establish the multihop session is set to 0 seconds. The **no** form of the command removes the time delay for the multihop session.

The *seconds* value is the time delay in seconds. You can specify a value between 0 and 600 seconds. The default value is 0 seconds.

Displaying MPLS BFD information

You can display the following information about an LSP BFD configuration:

- BFD Application Information
- BFD MPLS Information
- Detailed BFD MPLS Information
- MPLS BFD Global Configuration Information

You can also obtain MPLS BFD information using the **show bfd** command, as described in Displaying BFD information on page 855, and the **show mpls lsp** command, as described in "Displaying signalled LSP status information".

Displaying BFD application information

The following example illustrates the output from the **show bfd application** command.

```
device# show bfd application
Registered Protocols Count: 3
     Protocol  VRFID     Parameter HoldoverInterval
     isis      0         0         2
     ospf6     0         1         10
     ospf      0         0         5
```

This display shows the following information.

TABLE 116 Display of BFD application information

This field...	Displays...
Protocol	Specifies protocols registered to use BFD on the device. Possible values are mpls/0, ospf/0, ospf6/0, or isis_task/0
VRFID	The VRF ID of the protocol.
Parameter	The parameter value passed by the protocol during registration with BFD.
HoldoverInterval	The time by which the BFD session DOWN notification is delayed. If within that holdover time, the BFD session is UP then it is not notified of the BFD session flap.

Displaying BFD MPLS information

The following example shows output from the **show bfd mpls** command.

```
device# show bfd mpls
Total number of MPLS BFD sessions: 2
Session name                          State   Interface Holddown  Interval  RH
```

```
lsp1                                    UP      eth 1/2   3000000   1000000   Y
10.11.11.1/1/10.22.22.2                 UP      eth 1/2   3000000   1000000   Y
```

Syntax: show bfd mpls

This display shows the following information.

TABLE 117 Display of BFD MPLS information

This field...	Displays...
Total number of MPLS BFD Sessions	The number of BFD sessions that have been established on this device.
Session name	The name of the session: **For LSP Sessions –** the LSP name. **For RSVP Sessions –** the session-id which is displayed as IPv4 tunnel endpoint, tunnel ID, or extended tunnel ID.
State	The current state of the BFD session: Up Down A.DOWN - The administrative down state INIT - The Init state UNKNOWN - The current state is unknown
Interface	The logical port (physical or virtual port) on which the BFD packet is sent out. The physical port can be either an Ethernet, or VE-enabled interface. The VE interface ID is specified by the *vid* variable.
Holddown	The interval in microseconds after which the session will transition to the down state if no message is received.
Interval	The interval in microseconds at which the local device sends BFD messages to the remote peer.
RH	Heard from remote.

Displaying BFD MPLS detailed information

The following example shows a display of BFD MPLS detailed information as a result of the **show bfd mpls detail** command. To view BFD MPLS information for a single LSP or RSVP session, use the **show bfd mpls lsp** command.

> **NOTE**
> The **show bfd mpls lsp** command displays the same information as the **show bfd mpls rsvp-session** command.

```
device# show bfd mpls lsp lsp2
Session name                            State   Interface Holddown  Interval  RH
lsp2                                    UP      eth 1/2   3000000   1000000   Y
  Local: Disc: 3, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 500000, MinRxInterval: 500000, Multiplier: 3
  Remote: Disc: 3, Diag: 3, Demand: 1 Poll: 0
          MinTxInterval: 1000000, MinRxInterval: 1000000, Multiplier: 3
  Stats: RX: 305 TX: 305 SessionUpCount: 1 at SysUpTime: 0:0:4:46.200
  Session Uptime: 0:0:3:46.650, LastSessionDownTimestamp: 0:0:0:0.0
  Tx Port: eth 1/2, Rx Port: eth 1/2
```

Syntax: show bfd mpls [**detail** | **lsp** *name* | **rsvp-session** *src-addr dest-addr tnnl-id*]

This information shown in this display that is not defined in Displaying BFD MPLS information on page 876 is described in either Displaying BFD neighbor information on page 856 or Table 118.

TABLE 118 Display of BFD MPLS detail information

This field...	Displays...
Interface	The logical port (physical or virtual port) on which the BFD packet is sent out. The physical port can be either an Ethernet, or VE-enabled interface. The VE interface ID is specified by the *vid* variable.
Tx Port:	The physical port on which the BFD packet is sent. When applicable, the Tx Port field displays a VE interface ID specified by the *vid* variable.
Rx Port:	The physical port on which the BFD packet is received.

Displaying MPLS BFD global configuration information

You can use the **show mpls bfd** command to display the global configuration information for a device, as shown in the following.

```
device# show mpls bfd
MPLS BFD admin            = Enabled
Minimum TX interval       = 1000 msec
Minimum RX interval       = 1000 msec
Detection time multiplier = 3
```

Syntax: show mpls bfd

TABLE 119 Display of **BFD MPLS detail** command

This field...	Displays...
MPLS BFD admin	The global configuration state of MPLS BFD on the device: can be either Enabled or Disabled
Minimum TX interval	Desired Min Tx Interval - the minimum interval, in microseconds, the local system will use when transmitting BFD Control packets. The value zero is reserved.
Minimum RX interval	Required Min Rx Interval - the mimi num interval, in microseconds, between received BFD Control packets that this system is capable of supporting. If this value is zero, the transmitting system does not want the remote system to send any periodic BFD Control packets.
Detection time multiplier	The number of times in a single sequence this device waits to receive a BFD message from the peer before determining that the connection to that peer is not operational.