



BCSPNE in a Nutshell Study Guide for Exam 150-710



Brocade University

Revision 1111

Corporate Headquarters - San Jose, CA USA

T: (408) 333-8000
info@brocade.com

European Headquarters - Geneva, Switzerland

T: +41 22 799 56 40
emea-info@brocade.com

Asia Pacific Headquarters - Singapore

T: +65-6538-4700
apac-info@brocade.com

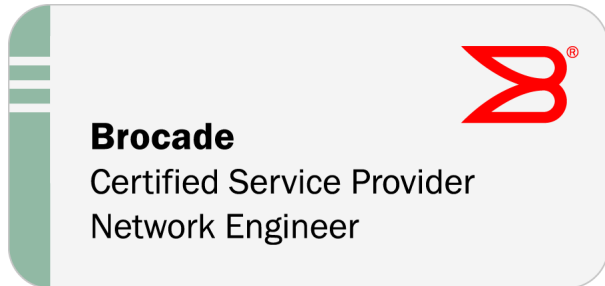
© 2011 Brocade Communications Systems, Inc. All Rights Reserved.

Brocade, the Brocade B-weave logo, Fabric OS, File Lifecycle Manager, MyView, Secure Fabric OS, SilkWorm, and StorageX are registered trademarks and the Brocade B-wing symbol and Tapestry are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. FICON is a registered trademark of IBM Corporation in the U.S. and other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

Revision 1111

BCSPNE in a Nutshell First Edition



Objective: The BCSPNE Nutshell guide is designed to help you prepare for the BCSPNE Certification, exam number 150-710.

Audience: The BCSPNE Nutshell self-study guide is intended for those who wish to undertake self-study or review activities before taking the actual BCSPNE exam. The BCSPNE guide is not intended as a substitute for classroom training or hands-on time with Brocade products.

How to make the most of the BCSPNE guide: The BCSPNE guide summarizes the key topics on the BCSPNE exam for you in an easy to use format. It is organized closely around the exam objectives. We suggest this guide be used in conjunction with our free online knowledge assessment test. To benefit from the BCSPNE guide, we strongly recommend you have successfully completed all prerequisite courses.

We hope you find this useful in your journey towards BCSPNE Certification, and we welcome your feedback by sending an email to jcannata@brocade.com.

Joe Cannata
Certification Manager

A handwritten signature in blue ink that reads "Joe Cannata".

Table of Contents

1 – Hardware

<i>Brocade MLXe</i>	1
<i>Managing Switch Fabric Modules</i>	3
<i>TCAM Partitioning and Usage</i>	4
<i>Interface Operations</i>	4
<i>Brocade 6910</i>	5

2 – Management Concepts

<i>Displaying and Modifying Default Settings for System Parameters</i>	6
<i>Enabling or Disabling Layer 2 Switching</i>	7
<i>sFlow</i>	7
<i>Management Module Redundancy</i>	8
<i>Access Control Lists (ACLs)</i>	9

3 – Quality of Service (QoS)

<i>Ingress Traffic Processing Through a Device</i>	11
<i>Configuring Ingress QoS Procedures</i>	12
<i>Egress Traffic Processing Exiting a Device</i>	13
<i>Displaying QoS Packet and Byte Counters</i>	13
<i>Control Traffic Prioritization</i>	13
<i>Scheduling Traffic for Forwarding</i>	14
<i>Configuring the Maximum Instantaneous Queue Size</i>	14
<i>Calculating the Values for WFQ Weight-based Traffic Scheduling</i>	15
<i>Egress Port and Priority Based Rate Shaping</i>	16
<i>Configuring Traffic Policing on Brocade Devices</i>	18

4 – Layer 2 Protocols

<i>Q-in-Q</i>	20
<i>Ethernet Service Instance (ESI) Overview</i>	21
<i>IEEE 802.1ad Provider Bridging (PB)</i>	22
<i>Provider Backbone Bridging (PBB)</i>	23
<i>Multi-Chassis Trunking (MCT)</i>	25
<i>Multi-Chassis Trunk (MCT) for VRRP or VRRP-E</i>	29
<i>Metro Ring Protocol (MRP)</i>	30
<i>Ethernet Ring Protection Protocol (ERP)</i>	33

5 – Layer 3 Protocols

VRRP-E	35
OSPF Concepts	37
Intermediate System-to-Intermediate System (IS-IS) Concepts	39
BGP Concepts	40
General IP Routing Protocol Concepts	43

6 – MPLS Concepts

How MPLS Works	45
Label Distribution Protocol (LDP)	48
MPLS Layer 2 VPNs	48
MPLS Layer 3 VPNs	51
Multi-VRF	51
BGP Extended Attributes for Layer 3 VPNs	52
Load Sharing for MPLS LAGs	53

7 – IP Multicast

Protocol Independent Multicast (PIM)	55
PIM Dense	55
PIM Sparse	55
Changing the Shortest Path Tree (SPT) Threshold	55
Concurrent Support for Multicast Routing and Snooping	56
Multicast Non-stop Routing	57
Multi-protocol Border Gateway Protocol (MBGP)	57

8 – IPv6

IPv6	58
IPv6 Multicast Addresses	59
Router ID in IPv6-only Networks	59
IPv6 Over IPv4 Tunnels	60
Configuring a Static IPv6 Route	60

9 – Monitoring, Maintenance, and Troubleshooting

<i>Showing System Software</i>	61
<i>Showing CPU Statistics</i>	61
<i>Displaying Information for an Interface</i>	61
<i>LACP Trunking</i>	62
<i>Port Mirroring and Monitoring</i>	63
<i>Protecting Against Smurf Attacks</i>	64
<i>BGP Neighbor States</i>	65

Taking the Test

List of Figures

MLX Clos Fabric Architecture	1
Logic Flow of Ingress QoS Processing	12
Logic Flow of Egress QoS Processing	13
Calculating WFQ Weight Formula	15
Calculating WFQ Weight Example	15
Ethernet Service Instance (ESI) for VLAN Configuration	21
IEE 802.1ad PB Network	22
Customer, PB, and PBB Frame Formats	23
Multi-Chassis Trunk example	25
MRP Ring - Normal State	30
MRP Ring - Initial State	31
Label Switching in an MPLS Domain	46
Multi-VRF Network	51
IPv6 Address Format	58
Smurf Attack	64

List of Tables

Parameters with adjustable table sizes	6
Port-based rate shaping interval.....	16
Port configuration for IEEE 802.1ah and IEEE 802.1ad.....	24
MCT feature interaction matrix.....	27
Feature differences between G.8032 version 1 and 2.....	34
Default Administrative Distances	43

1 – Hardware

After reviewing this section be sure you can perform the following:

- Describe the NetIron hardware platforms

Brocade MLXe

The Brocade MLX Series of routers is the most powerful suite of IPv4/IPv6/MPLS/Multi-VRF switching routers in the industry. They are a cost-efficient solution that is purpose-built to handle the most demanding applications with non-blocking, wire-speed performance. The Brocade MLX has a robust system architecture, a versatile feature set, and is available in four different sizes, making it capable of scaling from the edge to the core.

Scalable Clos Fabric Architecture

The Brocade MLX Series uses a Clos fabric architecture, which provides a high level of scalability, redundancy, and performance. As shown in Figure 1, there are multiple high-speed Switch Fabric Modules (hSFMs) in the system. A switch fabric module has multiple fabric elements, each of which has multiple connections to every interface slot.

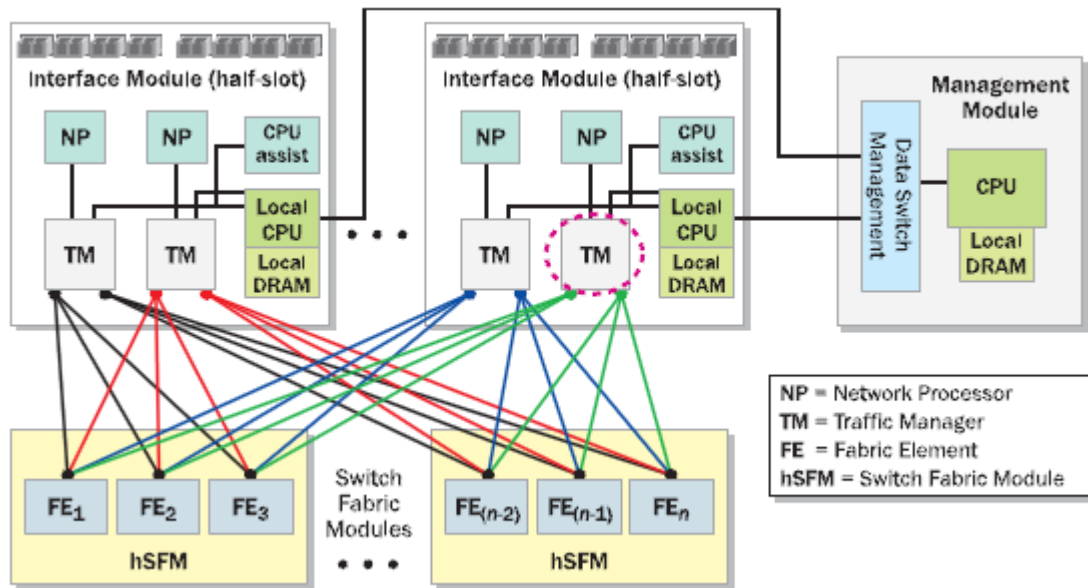


Figure 1: MLX Clos Fabric Architecture

The Clos architecture uses data striping technology to ensure optimal utilization of fabric interconnects. This mechanism always distributes the load equally across all available links between the input and output interface modules. By using fixed-size cells to transport packets across the switch fabric, the switching architecture of the Brocade MLX ensures predictable performance with very low and deterministic latency and jitter for any packet size. In addition, the Brocade MLX offers a “Turbo Mode” that increases switch fabric efficiency by using variable-size cells to transport packets. The presence of multiple switching paths between the input and output interface modules provides an additional level of redundancy.

These are some of the advantages of a Clos architecture over a traditional architecture:

- Common architecture across the product family because the same fabric elements are used on all chassis of the Brocade MLX Series. This demonstrates the superior scalability of the architecture from a small 4-slot system to a large 32-slot system.
- No head-of-line blocking at any point, irrespective of traffic pattern, packet size or type of traffic.
- Optimal utilization of switch fabric resources at all times. The data striping capability ensures that there is fair utilization of the switch fabric elements at all times without overloading of any single switch fabric element.
- “Intra-SFM” redundancy. An hSFM can withstand the failure of some of the fabric elements and still continue to operate with the remaining fabric elements. This unique capability provides a very high level of redundancy even within an hSFM.
- Exceptional high availability. The Brocade MLX supports redundant hSFMs. This allows the Brocade MLX to gracefully adapt to the failure of multiple switch fabric elements. Moreover, because there are multiple fabric elements within an hSFM, the failure of a fabric element does not bring down the entire hSFM.

Switch Fabric Modules

Switch fabric modules switch packets from one interface module to another. Brocade MLX Series and Brocade NetIron XMR routers can be configured with multiple switch fabric modules as described:

- 4-slot router: Accommodates three switch fabric modules (two required and one redundant) for a fully-loaded system. Ships with two switch fabric modules. You must purchase an additional switch fabric module to equip your router for redundancy.
- 8-slot router: Accommodates three switch fabric modules (two required and one redundant) for a fully-loaded system. Ships with two switch fabric modules. You must purchase an additional switch fabric module to equip your router for redundancy.
- 16-slot router: Accommodates four switch fabric modules (three required and one redundant) for a fully-loaded system. Ships with three switch fabric modules. You must purchase an additional switch fabric module to equip your router for redundancy.
- 32-slot router: Accommodates and ships with eight switch fabric modules.

High-speed Switch Fabric Modules

- Gen-2 high-speed fabric (hSFM) modules support wire-speed forwarding for all packet sizes, including jumbo frames
- hSFM modules are supported on Brocade MLX Series and Brocade NetIron XMR routers, and are interoperable with all existing interface module
- hSFM modules are hot-swappable, which means you can install or replace them while the system is powered up and running
- hSFM modules can operate in normal mode or turbo mode but will boot in turbo mode only if all active interface modules are Gen-2 module



Note

Gen-1 switch fabric modules and Gen-2 hSFMs are not compatible and will not operate together in the same device.

Managing Switch Fabric Modules

High-speed Switch Fabric Modules can operate in Normal mode or Turbo mode. Standard switch fabric modules (SFM) can only operate in Normal mode. The hSFM module is classified as a G2 module and SFM module is classified as a G1 module.

When operating in Normal mode, the system uses fixed size cells across the backplane. When operating in Turbo mode, the system uses variable size cells across the backplane. Turbo mode provides higher performance since it is a more efficient mechanism of sending cells across the backplane.

The system selects the operating mode for switch fabric modules at startup, or when the first switch fabric or interface module is installed. The system uses this mode for all modules that are subsequently installed. hSFM modules will boot in Turbo mode only if all active interface modules are G2 modules. In a chassis loaded with G1 and G2 modules, the hSFM modules will default to Normal mode.

If the system fabric mode is changed to Normal mode from Turbo mode, or vice versa, the system will not change the current operating mode unless the chassis is reloaded.

The switch fabric modes have the following restrictions:

- The system blocks discovery of any standard switch fabric (G1) module if you have issued the **system-init block-g1-sfm** command.
- If the system is operating in Turbo mode, standard switch fabric modules (G1) and standard (G1) interface modules are automatically blocked.
- If there are any active G1 switch fabric modules, G2 interface modules are blocked.
- If there are any active G2 interface modules, G1 switch fabric modules are blocked.

TCAM Partitioning and Usage

Ternary Content Addressable Memory (TCAM) is a component of Brocade devices that facilitates hardware forwarding. As packets flow through the Brocade device from a given source to a given destination, the management processor records forwarding information about the flow in TCAM entries. A TCAM entry generally contains next-hop information, such as the outgoing port, the MAC address of the next-hop router, a VLAN tag, and so on. Once the Brocade device has this information in TCAM, packets with the same source and destination can be forwarded by hardware, bypassing the management processor, and speeding up forwarding time.

TCAM entries can contain Layer 2, Layer 3, or Layer 4 information. Each type of TCAM entry has its own format:

- Layer 2 TCAM entries contain destination MAC information and deal with 802.1p (priority) and VLAN information.
- Layer 3 TCAM entries contain destination IP information.
- Layer 4 TCAM entries contain destination IP, destination TCP/UDP port, source IP, and source TCP/UDP port information.

When a Brocade device is initialized, the software partitions the available TCAM into segments for Layer 2, Layer 3, or Layer 4 information. The percentage of TCAM devoted to each type of TCAM entry is determined by a user-defined profile, if configured. Otherwise it is determined by the default TCAM profile.

Interface Operations

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.

Brocade 6910

The Brocade 6910 Ethernet Access Switch provides a flexible, easy-to-deploy solution to address service provider challenges.

Optimized for pure-play Ethernet access within metro areas, the Brocade 6910 supports applications for mobile backhaul, business services, and residential triple-play offerings. In addition, it

offers an energy-efficient and small form-factor solution that extends wire-speed Ethernet services to the network edge without compromising performance.



The Brocade 6910 is a carrier-grade access switch that meets the requirements for non-stop networking with Layer 2 services. The platform provides dual-power input for redundancy, and has been architected with a state-of-the-art hardware design and field-proven operating system for reliability and resiliency. It combines the low cost and high capacity of Ethernet with the reliability, management, and service quality needed for mission-critical applications. With its small form factor, fanless design, and temperature-hardened specifications, the Brocade 6910 is ideal for a variety of physical environments that support enterprise customers, managed service providers, Multi-Tenant Unit (MTU)/Multi-Dwelling Unit (MDU) deployments, and wireless backhaul providers.

Highlights

- Enables enterprise, wireless backhaul, and residential Fiber-To-The-Home (FTTH) services in a single Carrier Ethernet access switch
- Supports comprehensive Quality of Service (QoS) for multi-tiered Service Level Agreement (SLA) offerings
- Helps ensure high reliability with redundant power supplies and proactive service monitoring and management for fast resolution of performance issues
- Provides a single, cost-effective, and easy-to-manage platform
- Enables profitable Ethernet services to create new revenue streams

2 – Management Concepts

After reviewing this section be sure you can perform the following:

- Describe management features
- Demonstrate knowledge of security concepts

Displaying and Modifying Default Settings for System Parameters

The Brocade MLX has default table sizes for the following parameters. The table sizes determine the maximum number of entries the tables can hold. Individual table sizes can be adjusted to accommodate your configuration needs:

TABLE 1 Parameters with adjustable table sizes

MAC address entries	IPv6 Multicast routes
VLANs supported on a system	IPv6 PIM mcache
Virtual interfaces	Layer 4 sessions supported
Spanning tree instances	Number of VPLS's
RSTP instances	VPLS MAC entries
IP cache size	VRF routes
ARP entries	IPv6 cache
IP routes	IPv6 routes
IP ACL filter entries	Number of tunnels
L2 ACL entries per ACL table	Number of LAGs
Size for management port ACL	Configuration file size
IP subnets per port and per device	

The tables you can configure as well the defaults and valid ranges for each table differ depending on the Brocade device you are configuring.



Note

Changing the table size for a parameter reconfigures the device's memory. Whenever you reconfigure the memory on a Brocade device, you must save the change to the startup configuration file, then reload the software to place the change into effect.

Enabling or Disabling Layer 2 Switching

By default, Brocade NetIron devices support routing over Layer 2 switching. You can enable Layer 2 switching globally or on individual port using the **no route-only** command.



Note

On the Brocade NetIron CES and Brocade NetIron CER, the **route-only** command should not be configured on untagged MPLS uplinks when using it for VPLS or VLL. Otherwise, incoming VPLS or VLL traffic is dropped.

Always perform a reload after removing a route-only config or enabling route-only. Removing or enabling the route-only option without a reload will cause multicast issues.

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-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.

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 odd power of 2 to obtain the actual rate. This value becomes the actual sampling rate. For example, if the configured sampling rate is 1000, then the actual rate is 2048; and the hardware samples 1 in 2048 packets.

To change the default (global) sampling rate, enter a command such as the following at the global CONFIG level of the CLI.

```
Brocade(config)# sflow sample 2048
```

Syntax: [no] sflow sample <num>

The <num> parameter specifies the average number of packets from which each sample will be taken. The sampling rate you configure is the actual sampling rate. Valid values are 512 – 2147483648. The default is 2048.

Management Module Redundancy

When you apply power to or reload 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.

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

- The flash memory on the active management module
- A PCMCIA flash card in a PCMCIA slot on the active management module.

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

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

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

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

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.

Access Control Lists (ACLs)

ACLs are used to filter traffic based on the information in the IP packet header. You can use IPv4 ACLs to provide input to other features such as route maps, distribution lists, rate limiting, and BGP. When you use an ACL this way, use permit statements in the ACL to specify the traffic that you want to send to the other feature. If you use deny statements, the traffic specified by the deny statements is not supplied to the other feature.

Numbered and Named ACLs

When you configure IPv4 ACLs, you can refer to the ACL by a numeric ID or by an alphanumeric name. The commands to configure numbered ACLs are different from the commands for named ACLs:

- If you refer to the ACL by a numeric ID, you can use 1 – 99 for a standard ACL or 100 – 199 for an extended ACL. This document refers to this ACL as *numbered ACL*.
- If you refer to the ACL by a name, you specify whether the ACL is a standard ACL or an extended ACL, then specify the name. This document refers to this ACL type as *named ACL*.

You can configure up to 99 standard numbered IP ACLs and 100 extended numbered IP ACLs. You also can configure up to 100 standard named ACLs and 500 extended named ACLs.

Example: Standard numbered ACL

```
Router(config)# access-list 1 deny 209.157.25.0/24
Router(config)# access-list 1 deny host 209.157.29.12 log
Router(config)# access-list 1 deny host IPHost1 log
Router(config)# access-list 1 permit any
Router(config)# interface ethernet 1/1
Router(config-if-1/1)# ip access-group 1 in
```

In the example above, packets received on interface e1/1 will be examined and the following actions will take place:

- Packets from any host on the 209.157.25.0/24 subnet will be dropped
- Packets from the host with IP address 209.157.29.12, will be dropped and logged
- Packets from the device with a hostname of “IPHost1” will be dropped and logged
- All other packets will be allowed

Example: Extended numbered ACL

```
Router(config)# access-list 102 deny icmp 209.157.25.0/24 any echo
Router(config)# access-list 102 deny igmp any host 209.157.29.12 log
Router(config)# access-list 102 permit ip any any
Router(config)# interface ethernet 1/1
Router(config-if-1/1)# ip access-group 102 in
```

In the example above, packets received on interface e1/1 will be examined and the following actions will take place:

- ICMP echo packets **from** any host on the 209.157.25.0/24 subnet **to** any other host will be dropped
- IGMP packets **from** any host **to** the host with IP address 209.157.29.12, will be dropped and logged
- All other packets will be allowed

Configuring Standard or Extended Named ACLs

The commands for configuring named ACL entries are different from the commands for configuring numbered ACL entries. The command to configure a numbered ACL is **access-list**. The command for configuring a named ACL is **ip access-list**. In addition, when you configure a numbered ACL entry, you specify all the command parameters on the same command. When you configure a named ACL, you specify the ACL type (standard or extended) and the ACL name with one command, which places you in the configuration level for that ACL. Once you enter the configuration level for the ACL, the command syntax is the same as the syntax for numbered ACLs.

Example: Standard named ACL

```
Router(config)# ip access-list standard TestSTD
Router(config-std-nacl)# deny 209.157.25.0/24
Router(config-std-nacl)# deny host 209.157.29.12 log
Router(config-std-nacl)# deny host IPHost1 log
Router(config-std-nacl)# permit any
Router(config)# interface ethernet 1/1
Router(config-if-1/1)# ip access-group TestSTD in
```

In the example above, packets received on interface e1/1 will be examined and the following actions will take place:

- Packets from any host on the 209.157.25.0/24 subnet will be dropped
- Packets from the host with IP address 209.157.29.12, will be dropped and logged
- Packets from the device with a hostname of "IPHost1" will be dropped and logged
- All other packets will be allowed

Example: Extended named ACL

```
Router(config)# ip access-list extended TestEXT
Router(config-std-nacl)# deny icmp 209.157.25.0/24 any echo
Router(config-std-nacl)# deny igmp any host 209.157.29.12 log
Router(config-std-nacl)# permit ip any any
Router(config)# interface ethernet 1/1
Router(config-if-1/1)# ip access-group TestEXT in
```

In the example above, packets received on interface e1/1 will be examined and the following actions will take place:

- ICMP echo packets **from** any host on the 209.157.25.0/24 subnet **to** any other host will be dropped
- IGMP packets **from** any host **to** the host with IP address 209.157.29.12, will be dropped and logged
- All other packets will be allowed

3 – Quality of Service (QoS)

After reviewing this section be sure you can perform the following:

- Demonstrate knowledge of QoS
- Describe how to implement QoS

Ingress Traffic Processing Through a Device

The QoS operation on ingress traffic of a Brocade device involves reception and processing of packets based upon priority information contained within the packet. As the packets are processed through the device, there are several opportunities to influence the processing by configuration as described in the steps below.

1. Derive priority and drop precedence from the packets Priority Code Point (IEEE 802.1p) value. The Priority Code Point (PCP) is a 3-bit field within an IEEE 802.1Q tagged frame that is used to convey the priority of the frame. By using a mapping table, the 3-bit PCP field can be decoded to derive priority and drop precedence information.
2. Derive priority and drop precedence from the packets EXP value.
3. Derive priority and drop precedence from the packets DSCP value.
4. Merge or force the priorities described in steps 1 through 3.
5. Merge or force the priority and drop precedence value based on the value configured for the physical port.
6. Merge or force the priority value based on the value configured for the VLAN.
7. Merge or force the priority value based on an ACL look-up. This is used for setting a a specific priority for and L2, L3 or L4 traffic flow.

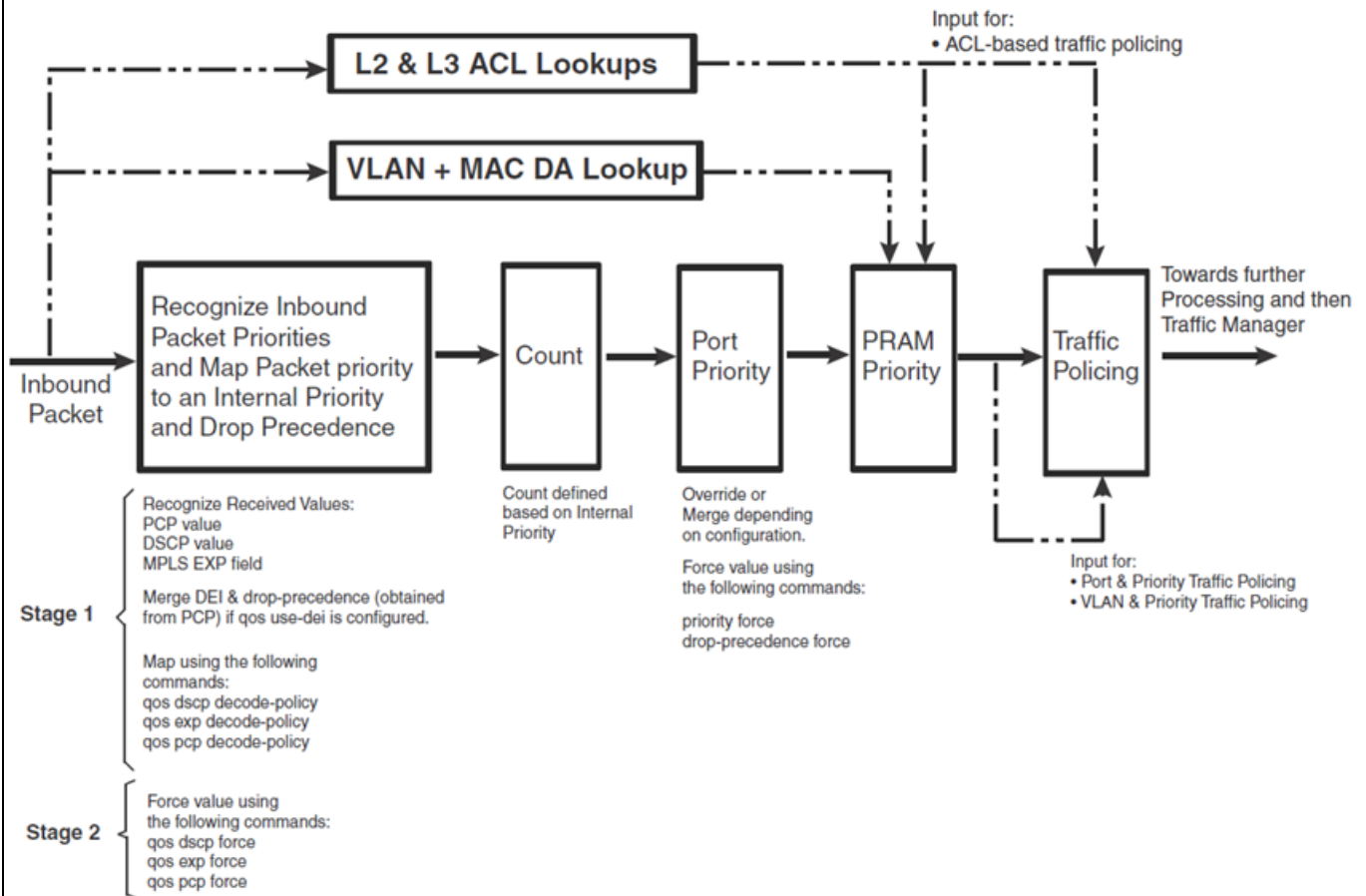


Figure 2: Logic Flow of Ingress QoS Processing

Configuring Ingress QoS Procedures

The following procedures are required to configure a Brocade device for ingress QoS processing:

- Creating ingress Decode Policy Maps** – Ingress Decode Policy Maps are created globally and are applied later either globally for all ports on a device or locally to specific port. To create an ingress Decode Policy Map, you must first enter the QoS mapping configuration level of the command interface using the **qos-mapping** command. Policy maps can use the following fields to assign priority or drop precedence values:
 - Differentiated Services Code Point (DSCP)
 - MPLS Experimental (EXP) bits
 - Priority Code Point (PCP)
- Binding ingress Decode Policy Maps** – To apply an ingress Policy Map other than the default, you must bind the ingress Policy Map either globally or to a specified interface.
- Configuring a Force priority** – Where there are multiple QoS values that can be used to determine the QoS level used on the device, the default policy is to determine the value used by performing a merge. Otherwise, you can specify a value that you want used from either the port or VLAN configured value or the DSCP, EXP or PCP values in the packets.

Egress Traffic Processing Exiting a Device

The QoS operation on egress traffic of a Brocade device involves marking packets as they leave a device on the egress port. As the packets are prepared to exit the device you can set the PCP, DSCP, and EXP values in the packet headers. This process is described in [Figure 3](#).

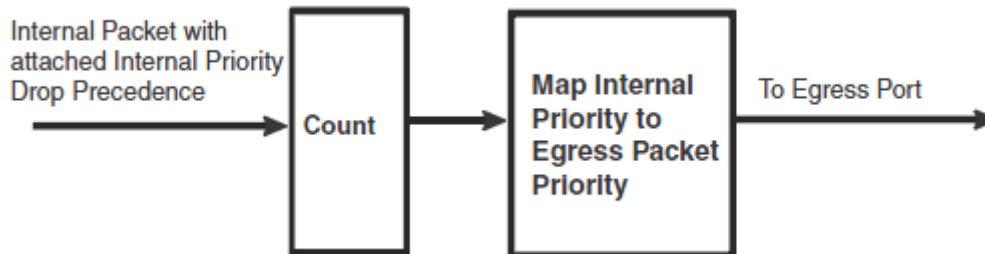


Figure 3: Logic Flow of Egress QoS Processing

Displaying QoS Packet and Byte Counters

You can enable the collection of statistics for ingress and egress packet priorities using the **enable-qos-statistics** command. Once the collection of statistics is enabled, the **show np qos statistics** command can be used to display a count of the packet priorities of ingress and egress packets as shown in the following.

```

Brocade# show np qos statistics eth 1/1
Port 1/1
  Ingress counters:
    COS 0: packets 0          bytes 0
    COS 1: packets 0          bytes 0
    COS 2: packets 0          bytes 0
    COS 3: packets 0          bytes 0
    COS 4: packets 0          bytes 0
    COS 5: packets 0          bytes 0
    COS 6: packets 0          bytes 0
    COS 7: packets 1122084909 bytes 134650189080
  Egress counters:
    COS 0: packets 0          bytes 0
    COS 1: packets 0          bytes 0
    COS 2: packets 0          bytes 0
    COS 3: packets 0          bytes 0
    COS 4: packets 4056756685 bytes 486810801752
    COS 5: packets 0          bytes 0
    COS 6: packets 0          bytes 0
    COS 7: packets 453        bytes 49490
  
```

Control Traffic Prioritization

The Traffic Manager (TM) allows prioritization and scheduling of packets destined for the CPU to guarantee optimal control packet processing and to reduce protocol flapping. The TM achieves physical separation of CPU-bound data and control packets. The hierarchical structure supports four sets of eight priority queues.

Scheduling Traffic for Forwarding

If the traffic being processed by a Brocade device is within the capacity of the device, all traffic is forwarded as received. Once we reach the point where the device is bandwidth constrained, it becomes subject to drop priority or traffic scheduling.

The Brocade devices classify packets into one of eight internal priorities. Traffic scheduling allows you to selectively forward traffic according to the forwarding queue that is mapped to according to one of the following schemes:

- **Strict priority-based scheduling** – This scheme guarantees that higher-priority traffic is always serviced before lower priority traffic. The disadvantage of strict priority-based scheduling is that lower-priority traffic can be starved of any access.
- **WFQ weight-based traffic scheduling** – With WFQ destination-based scheduling enabled, some weight-based bandwidth is allocated to all queues. With this scheme, the configured weight distribution is guaranteed across all traffic leaving an egress port and an input port is guaranteed allocation in relationship to the configured weight distribution.
- **Mixed strict priority and weight-based scheduling** – This scheme provides a mixture of strict priority for the three highest priority queues and WFQ for the remaining priority queues.

Configuring the Maximum Instantaneous Queue Size

You can set the maximum size to which a queue is allowed to grow. Packets that cause the queue to grow beyond this setting are unconditionally dropped. To set the maximum instantaneous queue size for queues with a queue type of 1 to 32000 KBytes, use the following command.

```
Brocade(config)# qos queue-type 1 max-queue-size 32
```

Syntax: [no] qos queue-type <queue-number> max-queue-size <max-queue>

The <queue-type> variable is the number of the forwarding queue type that you want to configure the instantaneous-queue-size parameter for. There are eight forwarding queue types on Brocade devices. They are numbered 0 to 7.

The <max-queue> variable is the maximum size to which a queue is allowed to grow. It is defined in Kbytes.

Calculating the Values for WFQ Weight-based Traffic Scheduling

Weighted Fair Queueing (WFQ) scheduling is configured to be a percentage of available bandwidth using the following formula:

$$\text{Weight of } q(x) = \frac{q(x)}{q0 + q1 + q2 + q3 + q4 + q5 + q6 + q7}$$

Figure 4: Calculating WFQ Weight Formula

Where $q(x)$ = The value of the queue that you want to determine the weight for. It can be the value of any queue (0 - 7).

$q0 - q7$ = the assigned values of the eight queues.

Weight of $q(x)$ = the calculated weight as a percentage of the port's total bandwidth.

For example if you assign the following values to queues 0 to 7:

- Queue 0 = 10, Queue 1 = 15, Queue 2 = 20, Queue 3 = 25, Queue 4 = 30, Queue 5 = 35, Queue 6 = 40, and Queue 7 = 45,

To determine the weight of $q3$, use the following formula:

$$\text{Weight of } q3 = \frac{25}{10 + 15 + 20 + 25 + 30 + 35 + 40 + 45}$$

Figure 5: Calculating WFQ Weight Example

The weight of $q3$ is 11.4%. Consequently, $q3$ will get 11.4% of the port's total bandwidth.

The values of the remaining queues are calculated to be the following: $q7 = 20.5\%$, $q6 = 18.2\%$, $q5 = 15.9\%$, $q4 = 13.6\%$, $q3 = 11.4\%$, $q2 = 9.1\%$, $q1 = 6.8\%$, and $q0 = 4.5\%$

Configuring WFQ Weight-based Traffic Scheduling

To configure WFQ weight-based scheduling use a command such as the following.

```
Brocade(config)# interface ethernet 1/1
Brocade(config-if-e1000-1/1)# qos scheduler weighted 5 10 15 20 30 15 5 10
```

Syntax: `qos scheduler weighted <queue7-weight> <queue6-weight> <queue5-weight> <queue4-weight> <queue3-weight> <queue2-weight> <queue1-weight> <queue0-weight>`

The `<queuex-weight>` variable is used to set the relative value for calculating each queue’s bandwidth allocation.

The acceptable range for `<queuex-weight>` variables is 1-128.

Egress Port and Priority Based Rate Shaping

Rate shaping is a mechanism to smooth out the variations in traffic above a certain rate. The primary difference between rate shaping and rate limiting is that in rate limiting, traffic exceeding a certain threshold is dropped. In rate shaping, the traffic that exceeds a threshold is buffered so that the output from the buffer follows a more uniform pattern. Rate shaping is useful when burstiness in the source stream needs to be smoothed out and a more uniform traffic flow is expected at the destination.



Note

Because excess traffic is buffered, rate shaping must be used with caution. In general, it is not advisable to rate shape delay-sensitive traffic.

Brocade devices support egress rate shaping. Egress rate shaping is supported per port or for each priority queue on a specified port.

Configuring Port-based Rate Shaping

When setting rate shaping for a port, you can limit the amount of bandwidth available on a port within the limits of the port’s rated capacity. Within that capacity, you can set the bandwidth at increments within the ranges described in [Table 2](#).

TABLE 2 Port-based rate shaping interval

Range	Increment supported within the range
0 - 10M	8,333
10M - < 100M	20,833
100M - < 1G	208,333
1G - 10G	2,083,333



Note

The egress rate shaping burst size for a port-based shaper is 10,000 bytes.

These limits provide a minimum and maximum rate that the port can be set to. They also provide the increments at which the port capacity can be set. In operation, you can set any number between the minimum and maximum values. The device will automatically round-up the value to the next higher increment.

For example, if you set the rate of a 10G port to 2,000,000,000, the actual rate would be 2,002,083,173. This is because it is the next highest increment above 2,000,000,000.

To set a 10 Gbps port to the incremental port capacity over 2 Gbps, use the following command.

```
Brocade(config)# interface ethernet 2/2  
Brocade(config-if-e10000-2/2)# qos shaper 2000000000
```

Syntax: [no] **qos shaper** <rate>

The <rate> variable sets the rate you want to set for the port within the limits available as described in [Table 2 on page 16](#). The rate is set in bps.

Configuring Traffic Policing on Brocade Devices

The following sections show examples of how to configure each traffic policing policy type.

Configuring a Policy Map

To configure a policy map, enter a command such as the following.

```
Brocade(config)# policy-map map1 cir 1000000 cbs 2000000 eir 1000000
ebs 2000000 excess-dp 2 excess-dscp 37
```

The command configures the traffic policing policy map map1 to limit CIR rate to 1000000 the CBS rate to 2000000, the EIR rate to 1000000 and the EBS to 2000000. In addition, traffic that exceeds the bandwidth available in the CIR bucket will have its packets drop precedence set to 2 and its DSCP set to 37. This command only creates a policy, it must be applied to one or more ports to be operational.

Syntax: [no] **policy-map** <map-name> **cir** <cir-rate> **cbs** <cbs-rate> [**eir** <eir-rate> **ebs** <ebs-rate> **excess-priority** <priority-num> [**excess-dscp** <dscp-num>]] | [**eir** <eir-rate> **ebs** <ebs-rate> **excess-dp** <dp-val> [**excess-dscp** <dscp-num>]]

The map-name variable is the name you will use to reference the policy map in traffic policing command. It can be a character string up to 64 characters long.

- **cir** - defines the value of the Committed Information Rate (CIR) as the rate defined in the <cir-rate> variable. Acceptable values are: 0 - 10000000000 bps in increments of 8,144 bps.
- **cbs** - defines the value of the Committed Burst Size (CBS) as the rate defined in the <cbs-rate> variable. Acceptable values are: 1250 - 1250000000 bytes in increments of 1 byte.
- **eir** - defines the value of the Excess Information Rate (EIR) as the rate defined in the <eir-rate> variable. Acceptable values are: 0 - 10000000000 bps in increments of 8,144 bps.
- **ebs** - defines the value of the Excess Burst Size (EBS) as the rate defined in the <ebs-rate> variable. Acceptable values are: 1250 - 1250000000 bytes in increments of 1 byte.
- **excess-priority** - specifies that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket will have its packets priority queue set to the value set in the <priority-num> variable. Acceptable values for the <priority-num> are 0-7.
- **excess-dp** - specifies the Weighted Random Early Detection (WRED) drop precedence for traffic whose bandwidth requirements exceed what is available in the CIR bucket and is sent to the EIR bucket. Acceptable values for the <dp-value> are 0-3. Packets with a value of 0 are least likely to be dropped and packets with a value of 3 are most likely to be dropped.
- **excess-dscp** - specifies that traffic whose bandwidth requirements exceeds what is available in the CIR bucket and is sent to the EIR bucket will have its packets DSCP priority set to the value set in the <dscp-num> variable. Acceptable values for the <dscp-num> are 0-63. When this parameter is used together with the **excess-dp** parameter, the value set for bits 2:1 (zero-based) in the **excess-dscp** parameter must be equal to the value set for **excess-dp**.

Configuring Port-based Traffic Policing for Inbound and Outbound Ports

Port-based traffic policing limits the rate on an individual inbound or outbound physical port to a specified rate.

To configure port-based traffic policing policy for outbound ports, enter commands such as the following at the interface level.

```
Brocade(config)# interface ethernet 1/1  
Brocade(config-if-1/1)# rate-limit out 500000000 750000000
```

The commands configure a traffic policing policy for outbound traffic on port 1/1. The policy limits the average rate of all outbound traffic to 500 Mbps (megabits per second) with a maximum burst size of 750 MBps (megabytes per second) or 6 Gbps (gigabits per second).

To configure port based traffic policing policy through a policy map, enter a command such as the following.

```
Brocade(config)# interface ethernet 1/1  
Brocade(config-if-1/1)# rate-limit input policy-map map1
```

The commands configure a traffic policing policy for inbound traffic on port 1/1. The policy references the policy map map1 for rate limiting policy parameters.

The complete syntax for configuring a port-based traffic policing policy is:

Syntax: [no] **rate-limit** [in | out] [<average-rate> <maximum-burst> | **policy-map** <map-name>]

The **in** parameter applies the policy to traffic on inbound ports.

The **out** parameter applies the policy to traffic on outbound ports.

Only one inbound and one outbound port-based traffic policing policy can be applied to a port.

4 – Layer 2 Protocols

After reviewing this section be sure you can perform the following:

- Describe Provider Backbone Bridging concepts
- Demonstrate knowledge of multi-chassis trunking
- Demonstrate knowledge of Metro Ring Protocol (MRP) and Ethernet Ring Protocol (ERP) features

Q-in-Q

Q-in-Q allows multiple VLAN headers to be inserted into a single frame, an essential capability for implementing Service Provider or Metro Ethernet network topologies.

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.
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.

Ethernet Service Instance (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 4096 VLANs can be configured across the entire network domain. With a Q-in-Q bridging, VLANs from the set of 4096 VLANs can be inter-connected across a provider network. While Ethernet Switching Instance (ESI) allows a carrier to provide transport services for different sets of 4096 VLANs for different customers, the provider network is still limited to using 4096 VLANs across all of the customers connected to a single box, as it is very difficult to configure and manage different sets of 4096 VLANs across the different ports within a single system.

Using an Ethernet Switching Instance (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 Layer 2 parameters that define properties of all of the elements contained within the instance.

In a simplified network shown in [Figure 6](#), customers A and B are connected to a Brocade NetIron CES device with each customer having a separate set of 4096 VLANs. One or more ESIs are created to hold these 4096 VLANs.

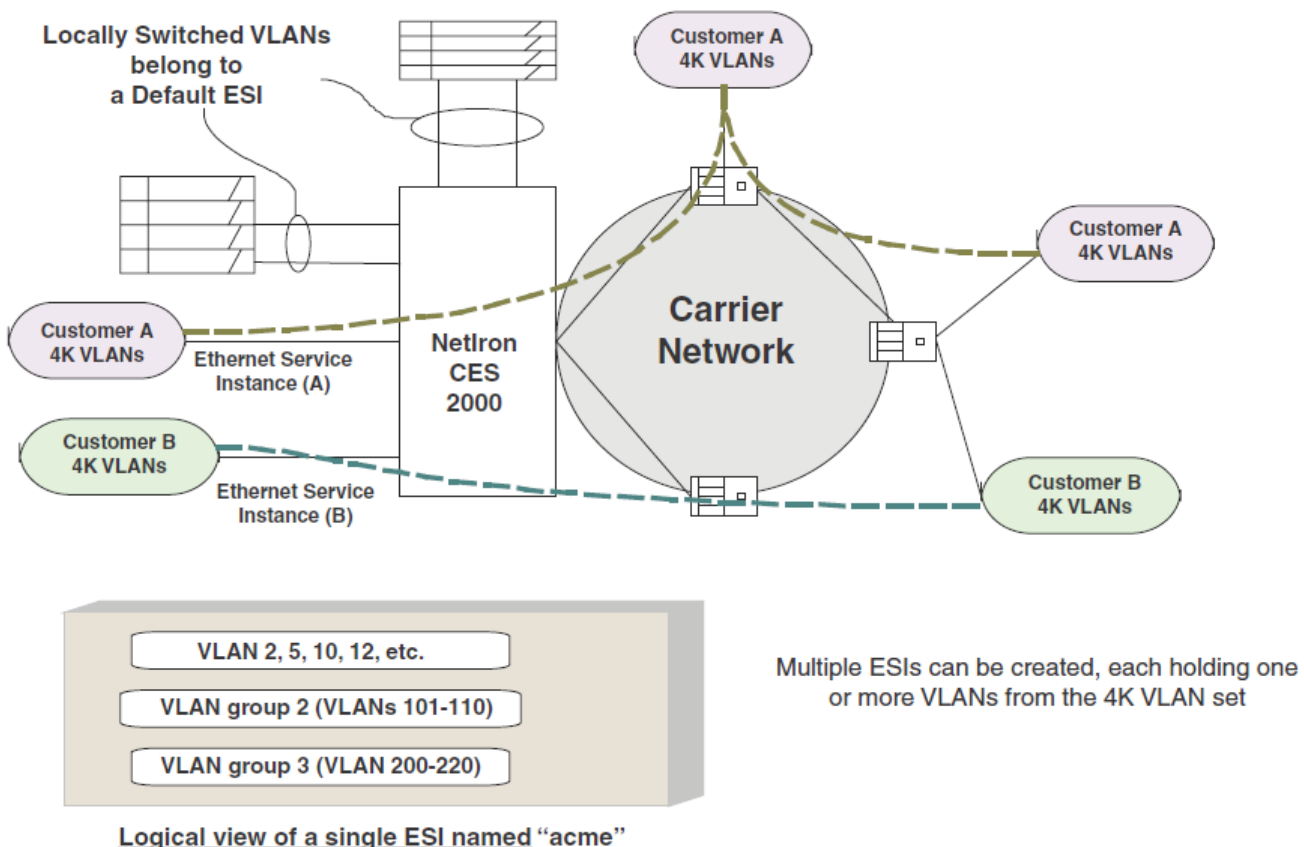


Figure 6: Ethernet Service Instance (ESI) for VLAN Configuration

IEEE 802.1ad Provider Bridging (PB)

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 7 provides an example of a PB network.

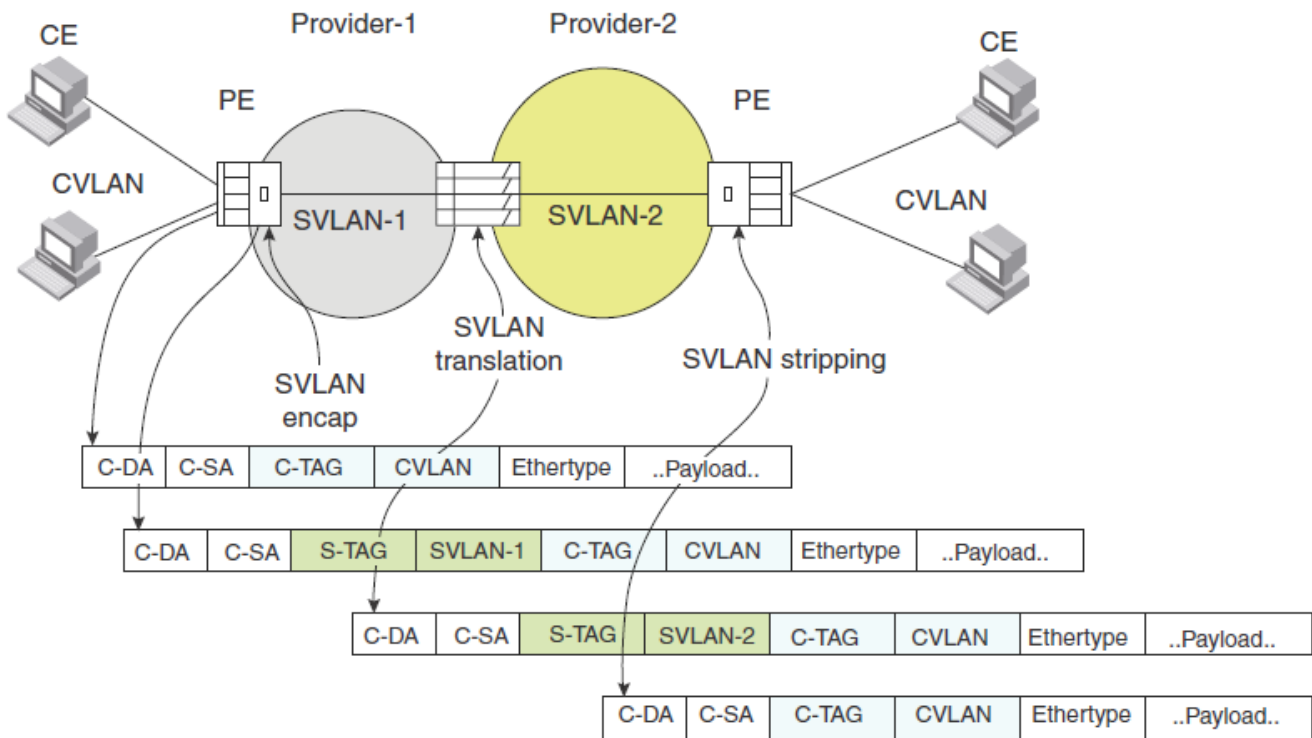


Figure 7: IEEE 802.1ad PB Network

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 4096 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).

Provider Backbone Bridging (PBB)

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 VLAN 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. 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, i.e., IEEE 802.1Q, which is known as defining “Q” frames.

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.

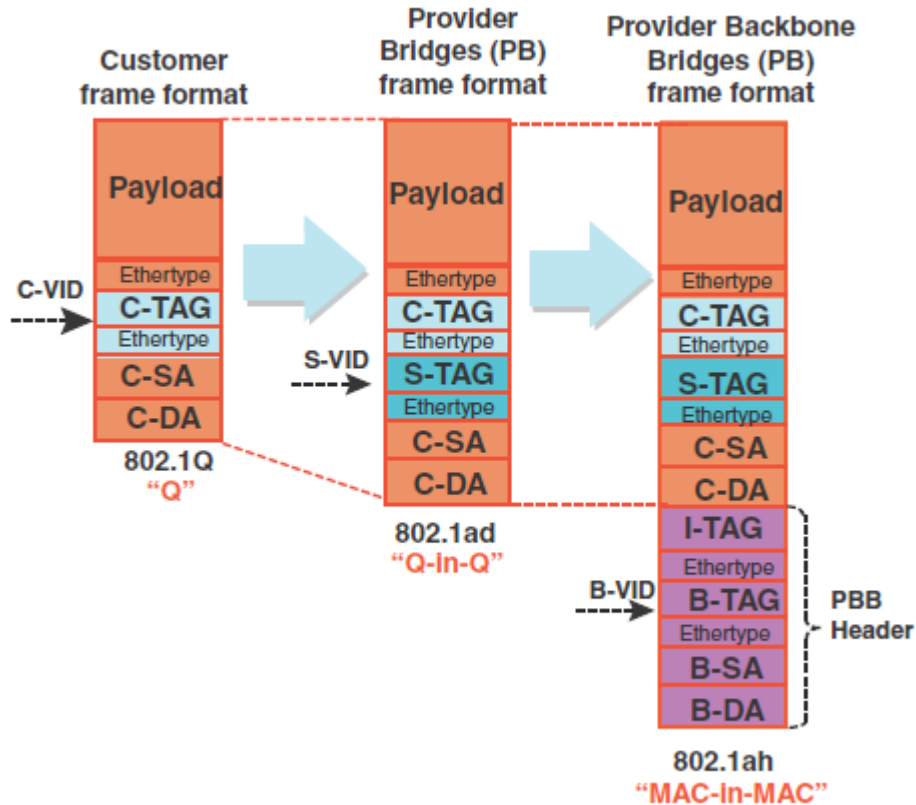


Figure 8: Customer, PB, and PBB Frame Formats

TABLE 3 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.	<ul style="list-style-type: none"> • CVLAN tag
PB_PN	Provider Network Port (IEEE 802.1ad)	<ul style="list-style-type: none"> • SVLAN tag • IEEE 802.1ad frame at this interface
PBB_BE	Backbone-Edge (IEEE 802.1ah)	<ul style="list-style-type: none"> • SVLAN tag • IEEE 802.1ad frame • This is the 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)	<ul style="list-style-type: none"> • BVLAN tag

Multi-Chassis Trunking (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, providing switch level redundancy.

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.

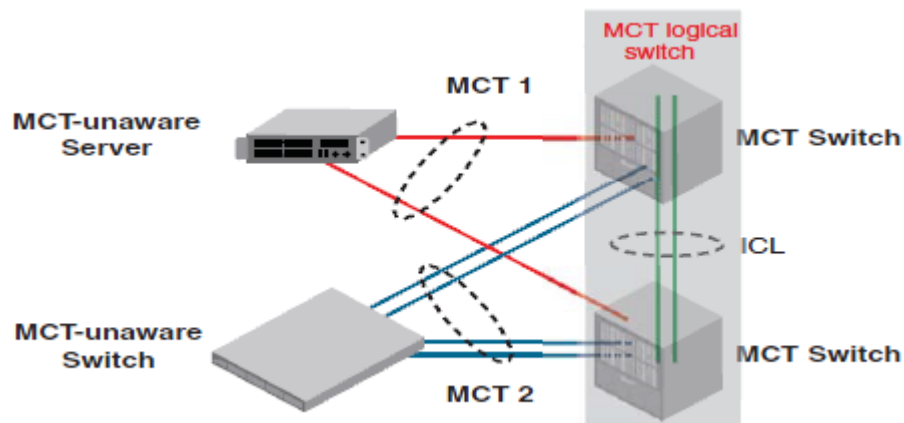


Figure 9: Multi-Chassis Trunk example

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 it 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 multi-port 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 MLX series, non-MCT VLANs can co-exist with MCT VLANs on the ICL. However, on the Brocade NetIron CES and Brocade NetIron CER, only MCT VLANs are carried over ICL.

- **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 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 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 protocol runs over this VLAN. The interface can be a single link or LAG port. If it is LAG port, it should be the primary port of the LAG. Note: 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 Source MAC with a MCT node.
- **MDUP:** MAC Database Update Protocol
- **CL:** Cluster Local MACs
- **CCL:** Cluster Client Local MACs
- **CR:** Cluster Remote MACs
- **CCR:** Cluster Client Remote MACs
- **CCRR:** Cluster Client RBridge Reachability
- **MDB:** Mac Data Base. The MDB can have multiple MAC entries for the same address
- **FDB:** Forwarding MAC Database. The FDM will have the best MAC only installed

MCT Feature Interaction

Use the following feature matrix when configuring MCT:

TABLE 4 MCT feature interaction matrix

Supported	Not Supported
LACP on both ICL and CCEP	MSTP, VSRP, RIP, OSPF, IS-IS, and BGP
VRRP on the CCEP	ESI VLANs on CCEP or ICL ports
MRP and MRP II supported with the restriction that ICL port cannot be the secondary port of the MRP ring	ESI VLANs on CCEP or ICL ports
Flooding features (VLAN CPU protection, multicast flooding etc.) on cluster VLANs	GRE is not supported on the ICL ve interfaces
UDLD as independent boxes	DAI on the CCEP ports
Link OAM as independent boxes	802.1ah on CCEP or ICL ports
802.1ag as independent boxes	VPLS on CCEP or ICL ports
ARP as independent boxes	VLL on CCEP or ICL ports
STP and RSTP	MPLS on CCE or ICL ports
L3 Routing - The IP address assignment is OK on CCEP ports for VRRP purpose. However, routing protocols would not be enabled on CCEP ports	MSTP
Port MAC Security on the node where it is programmed	Hitless Fail over is NOT supported on the Brocade 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 revalidated and programmed accordingly. Brocade recommends shutting down all the CCEP ports on the cluster node so that there is graceful failover and then hitless operation can be performed
802.1x on the node where it is programmed	Hitless Upgrade is NOT supported, on the Brocade 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 revalidated and programmed accordingly. Brocade recommends shutting down all the CCEP ports on the cluster node so that there is graceful failover and then hitless operation can be performed
Static MAC configuration - Static MACs are programmed on both local and remote peers as static entries	Multi-port MAC are not supported on ICL or CCEP ports. Configuration will be rejected when trying to configure multi-port MAC addresses with a port mask which contains either a CCEP port or ICL port and vice-versa on the Brocade MLXe, NetIron MLX and NetIron XMR
	Multi-port ARP will not be allowed on ICL or CCEP ports on the Brocade MLX series and NetIron XMR

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.

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.

```
Brocade(config)# cluster TOR 1
Syntax: [no] cluster <cluster-name> <cluster-id>
```

The *<cluster-name>* parameters specify the cluster name with a limit of 64 characters.

The *<cluster-id>* parameters specify the cluster ID (1-65535). The default is the device ID.

2. Configure the remote bridge ID cluster on the device so all clients for the device become members of the cluster.

```
Brocade(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

```
Brocade(config-cluster-TOR)# session-vlan 4090
Syntax: [no] session-vlan <vlan-d>
```

The *<vlan-id>* parameters specify the VLAN range. Possible values are 1 - 4090.

4. Specify the VLAN range on which cluster is operating. This would be the range for which there would be MAC synchronization. Multiple VLAN ranges would be supported for the configuration. Enter a command such as the following to create the member VLAN.

```
Brocade(config-cluster-TOR)# member-vlan 2
Syntax: [no] member-vlan <x> to <y>
```

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.

```
Brocade(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.

5. Specify the rbridge and ICL for the peers by entering a command such as the following.

```
Brocade(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.

6. 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.

```
Brocade(config-cluster-TOR) # deploy
```

Syntax: [no] **deploy**

Multi-Chassis Trunk (MCT) for VRRP or VRRP-E

One MCT Switch is the VRRP or VRRP-E Master Router and the Other MCT Switch is VRRP or VRRP-E Backup Router

The MCT switch that acts as backup router needs to ensure that packets sent to a VRRP-E virtual IP address can be L2 switched to the VRRP-E master router for forwarding. The MCT switch that acts as master router will sync the VRRP-E MAC to the other MCT switch that acts as backup router. Both data traffic and VRRP-E control traffic travel through the ICL unless the short-path forwarding feature is enabled.

L3 Traffic Forwarding from CEP Ports to CCEP Ports

Traffic destined to the CCEP ports from the client or CEP ports follow the normal IP routing on both master and backup routers. By default, the best route should not involve the ICL link. Only when the direct link from CEP ports to CCEP ports are down will the traffic be re-routed to pass through ICL link.

Both MCT Switches are VRRP or VRRP-E Backup Routers

Both MCT switches 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.

Under the VRRP-E VRID configuration level, use the short-path-forwarding command. 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>]

Metro Ring Protocol (MRP)

Brocade MRP is a proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in ring topologies. It is an alternative to Spanning Tree Protocol (STP) and is especially useful in Metropolitan Area Networks (MANs) where using 802.1D STP has the following drawbacks:

- 802.1D recommends a maximum bridge diameter of seven nodes with standard timers. MRP is capable of many more nodes than this.
- 802.1D has a slow reconvergence time, taking many seconds or even minutes. MRP can detect and heal a break in the ring in under one second.

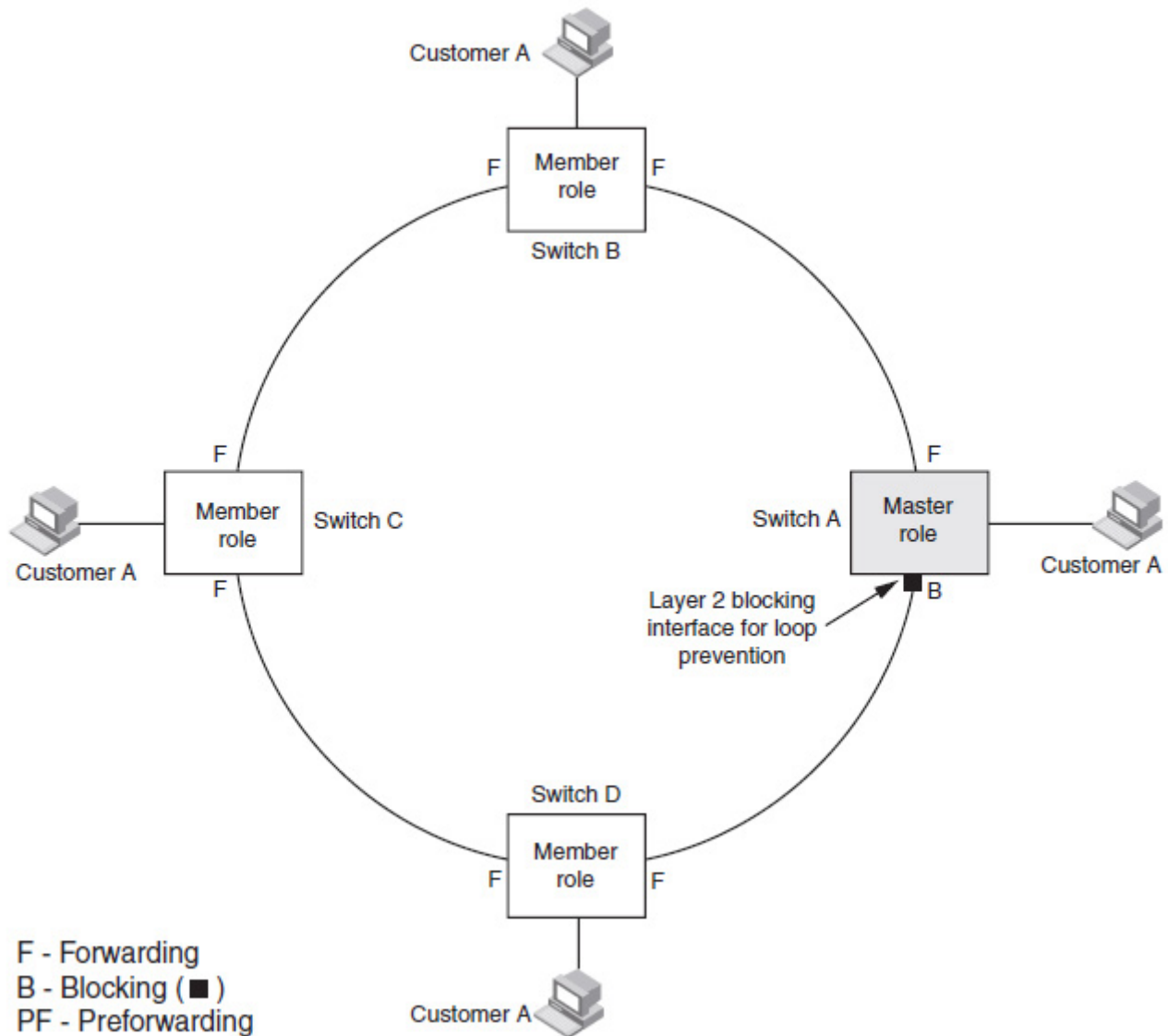


Figure 10: MRP Ring - Normal State

MRP Ring initialization

Figure 11 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).

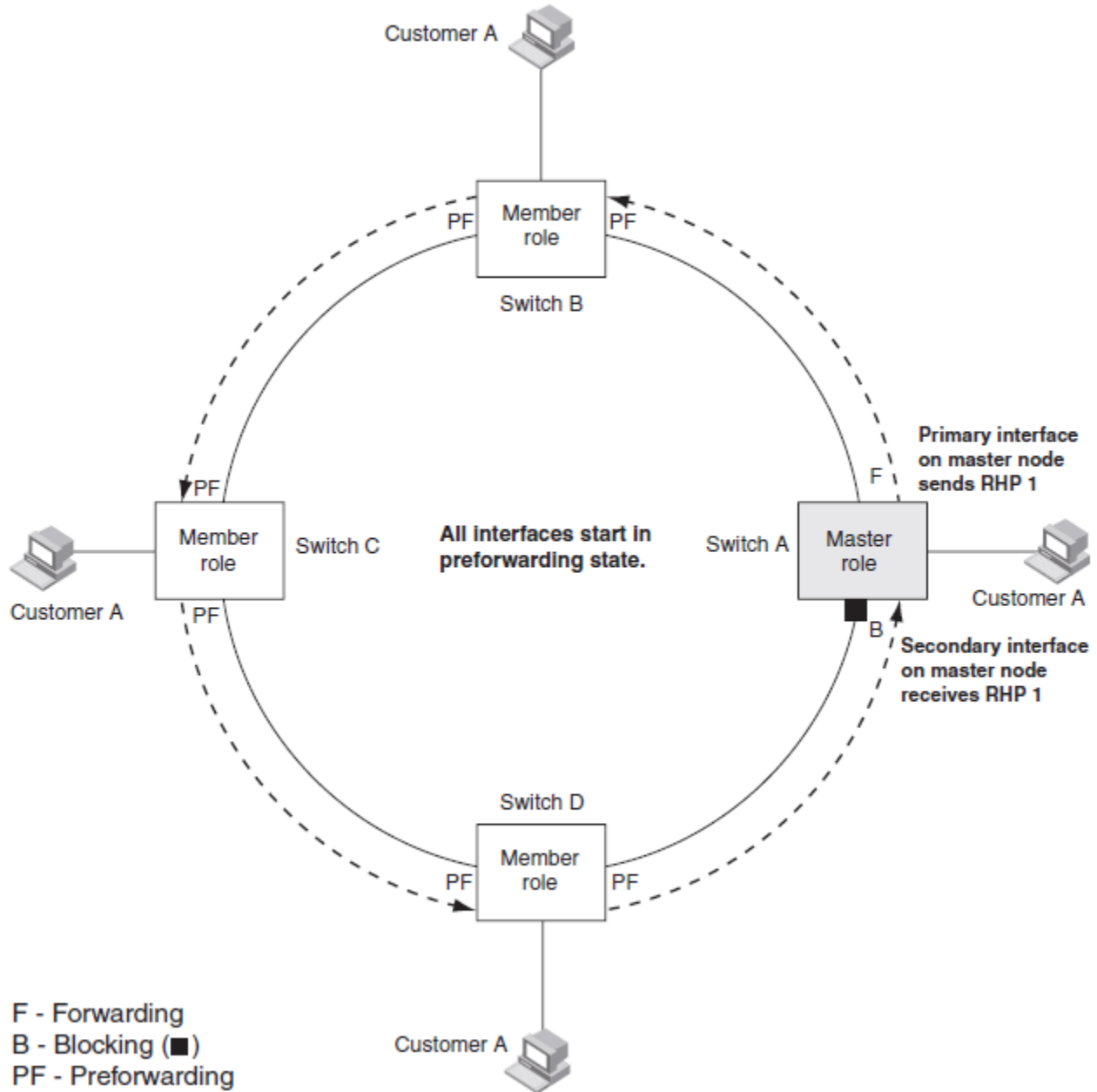


Figure 11: MRP Ring - Initial State

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 RHPs 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.

Configuring MRP

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.

```
Brocade(config)# vlan 2
Brocade(config-vlan-2)# metro-ring 1
Brocade(config-vlan-2-mrp-1)# name CustomerA
Brocade(config-vlan-2-mrp-1)# ring-interface ethernet 1/1 ethernet 1/2
Brocade(config-vlan-2-mrp-1)# enable
Brocade(config-vlan-2-mrp-1)# master
```

The ring interfaces are 1/1 and 1/2. The first interface listed will be allocated as the primary interface and the second will be allocated as the secondary interface. The primary interface initiates RHPs. The ring takes effect in vlan 2.

Syntax: [no] **metro-ring** <ring-id>

The <ring-id> parameter specifies the ring ID 1 - 255. Configure the same ring ID on each of the nodes in the ring.

Syntax: [no] **name** <string>

The <string> parameter specifies a name for the ring. The name is optional, but it can be up to 20 characters long and can include blank spaces. If you use a name that has blank spaces, enclose the name in double quotation marks (for example: "Customer A").

Syntax: [no] **master**

Configures this node as the master node for the ring. Enter this command only on one node in the ring. The node is a member (non-master) node by default.

Syntax: [no] **ring-interface ethernet** <primary-if> **ethernet** <secondary-if>

The ethernet <primary-if> parameter specifies the primary interface. On the master node, the primary interface originates RHPs. Ring control traffic will flow out of this interface by default. On member nodes the order in which you enter the interfaces does not matter as the secondary interface is determined by the receipt of RHP's from the master meaning the other interface defined in config becomes the primary. Once the ring is enabled the configuration entries on a member switch will reflect the ring direction no matter what order they are originally entered.

The ethernet <secondary-if> parameter specifies the secondary interface.

Syntax: [no] **enable**

The enable command enables the ring.

Tuning MRP Timers

To effectively tune MRP timers it is crucial to understand the association between the hello time and the preforwarding time.

Hello Time

This timer specifies the interval at which RHPs 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 RHPs being received. It also defines the amount of time an interface will take to move from preforwarding to forwarding without RHPs 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 RHPs are generated) will take one minute to heal.

Ethernet Ring Protection Protocol (ERP)

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.

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.

Setting the ITU-T G.8032 Version Number

You can configure the ERP configuration to use G.8032 version 1 or 2. The default value is version 2. [Table 5](#) lists the feature and MAC ID differences between versions 1 and 2.

TABLE 5 Feature differences between G.8032 version 1 and 2

Version	Supported ERP features	Treatment of MAC ID
1	<ul style="list-style-type: none"> • Signal Fail • Signal Fail recovery 	Always uses 01:19:A7:00:00:01 as the ERP ID in R-APS messages
2	<ul style="list-style-type: none"> • Signal Fail • Signal Fail recovery • Manual Switch • Forced Switch • Non-revertive • Interconnected rings • RPL configuration on non-RPL owner 	Allows use of the ERP ID for the last two bytes of the MAC ID (01:19:A7:00:00:erp-id)

5 – Layer 3 Protocols

After reviewing this section be sure you can perform the following:

- Describe VRRP-E operations
- Describe IP routing protocol concepts
- Describe advanced knowledge of BGP concepts

VRRP-E

VRRP-E is proprietary version of VRRP that overcomes limitations in the standard protocol. VRRP-E is similar to VRRP, but differs in the following respects:

Owners and Backups:

- VRRP-E does not use Owners. All routers are Backups for a given virtual router. The router with the highest priority becomes the Master. If there is a tie for highest priority, the router with the highest IP address becomes the Master. The elected Master owns the virtual IP address and answers ping and ARP requests and so on.

Master and Backups:

- VRRP-E – The Master and Backups are selected based on their priority. You can configure any of the Brocade devices to be the Master by giving it the highest priority. There is no Owner.

Virtual Router's IP address:

- VRRP-E requires only that the virtual router's IP address be in the same subnet as an interface configured on the VRID's interface. In fact, VRRP-E does not allow you to specify an IP address configured on the interface as the VRID IP address.

VRID's MAC Address:

- VRRP-E uses the interface's actual MAC address as the source MAC address. The virtual MAC address for IPv4 VRRP-E and IPv6 VRRP-E is 02-E0-52-*<hash-value>*-*<vrid>*, where *<hash-value>* is a two-octet hashed value for the IP address and *<vrid>* is the virtual router ID.



Note

You cannot reuse the same VRID across IPv4 VRRP-E and IPv6 VRRP-E, if they are in the same broadcast domain.

Hello packets:

- VRRP-E uses UDP to send Hello messages in IP multicast messages. The Hello packets use the interface's actual MAC address and IP address as the source addresses. The destination MAC address is 01-00-5E-00-00-02, and the destination IP address is 224.0.0.2 (the well-known IP multicast address for "all routers"). Both the source and destination UDP port number is 8888. VRRP messages are encapsulated in the data portion of the packet.

Track ports and track priority:

- VRRP-E reduces the priority of a VRRP-E interface by the amount of a tracked interface's priority if the tracked interface's link goes down. For example, if the VRRP-E interface's priority is 200 and a tracked interface with track priority 20 goes down, the software changes the VRRP-E interface's priority to 180. If another tracked interface goes down, the software reduces the VRID's priority again, by the amount of the tracked interface's track priority.

Configuration Rules and Feature Limitations for VRRP-E

Consider the following rules when configuring VRRP-E:

- The interfaces of all routers in a virtual router must be in the same IP subnet.
- The IP address assigned to the virtual router cannot be configured on any of the Brocade devices.
- The Hello interval must be set to the same value on all the Brocade devices.
- The Dead interval must be set to the same value on all the Brocade devices.
- The track priority for a virtual router must be lower than the VRRP-E priority.
- The same VRID must not be used across IPv6 VRRP-E and IPv4 VRRP-E if they are in the same broadcast domain.
- Hitless switchover is not supported.

Backup Preempt

By default, a Backup that has a higher priority than another Backup that has become the Master can preempt the Master, and take over the role of Master. If you want to prevent this behavior, disable preemption.

Preemption applies only to Backups and takes effect only when the Master has failed and a Backup has assumed ownership of the virtual router. The feature prevents a Backup with a higher priority from taking over as Master from another Backup that has a lower priority but has already become the Master of the virtual router.

VRRP-extended Slow Start

In a VRRP-E configuration, if a Master router goes down, the Backup router with the highest priority takes over after expiration of the dead interval. When the original Master router comes back up again, it takes over from the Backup router (which became the Master router when the original Master router went down). By default, this transition from Backup router back to Master router takes place after expiration of the dead interval.

You can configure the VRRP-E slow start timer feature, which causes a specified amount of time to elapse between the time the original Master router is restored and when it takes over from the Backup router (This range is currently set to between 1-60 seconds). This interval allows time for OSPF convergence when the Master is restored.

OSPF Concepts

Link State Cost

Each interface on which OSPF is enabled has a cost associated with it. The Layer 3 Switch advertises its interfaces and their costs to OSPF neighbors. For example, if an interface has an OSPF cost of ten, the Layer 3 Switch advertises the interface with a cost of ten to other OSPF routers.

By default, an interface's OSPF cost is based on the port speed of the interface. The cost is calculated by dividing the reference bandwidth by the port speed. The default reference bandwidth is 100 Mbps, which results in the following default costs:

- 10 Mbps port – 10
- All other port speeds – 1 (If the resulting cost is less than 1, the software rounds the cost up to 1.)

You can change the reference bandwidth, to change the costs calculated by the software. The software uses the following formula to calculate the cost:

Cost = reference-bandwidth/interface-speed

For 10 Gbps OSPF interfaces, in order to differentiate the costs between 100 Mbps, 1000 Mbps, and 10,000 Mbps interfaces, you can set the auto-cost reference bandwidth to 10000, whereby each slower link is given a higher cost, as follows:

- 10 Mbps port's cost = $10000/10 = 1000$
- 100 Mbps port's cost = $10000/100 = 100$
- 1000 Mbps port's cost = $10000/1000 = 10$
- 10000 Mbps port's cost = $10000/10000 = 1$

The bandwidth for interfaces that consist of more than one physical port is calculated as follows:

- Trunk group: The combined bandwidth of all the ports.
- Virtual interface: The combined bandwidth of all the ports in the port-based VLAN that contains the virtual interface.

The default reference bandwidth is 100 Mbps. You can change the reference bandwidth to a value from 1 – 4294967. If a change to the reference bandwidth results in a cost change to an interface, the Layer 3 Switch sends a link state update to update the costs of interfaces advertised by the Layer 3 Switch.

OSPF Hello Packet

All OSPFv2 routers send Hello messages to 224.0.0.5 (all OSPF routers on this subnet) to find neighbors. Certain items must match on both routers for them to become OSPF neighbors. These items are: subnet mask, area ID, Hello/Dead intervals, authentication password, and stub flag.

OSPFv2 areas

After OSPFv2 is enabled, you can assign OSPF areas. You can assign an IPv4 address or a number as the *area ID* for each area. The area ID is representative of all IP subnets on a device interface. Each device interface can support one area.

An area can be *normal*, a *stub*, or a *Not-So-Stubby Area (NSSA)*:

- **Normal** – OSPF routers within a normal area can send and receive external Link State Advertisements (LSAs).
- **Stub** – OSPF routers within a stub area cannot send or receive external LSAs. In addition, OSPF routers in a stub area must use a default route to the Area Border Router (ABR) or Autonomous System Boundary Router (ASBR) of the area to send traffic out of the area. By default, the device sends summary LSAs (type 3 LSAs) into stub areas.
- **NSSA** – The ASBR of an NSSA can import external route information into the area.
 - ASBRs redistribute (import) external routes into the NSSA as type 7 LSAs. Type 7 External LSAs are a special type of LSA generated only by ASBRs within an NSSA, and are flooded to all the routers within only that NSSA.
 - ABRs translate type 7 LSAs into type 5 External LSAs, which can then be flooded throughout the AS. You can configure address ranges on the ABR of an NSSA so that the ABR converts multiple type 7 External LSAs received from the NSSA into a single type-5 External LSA.

When an NSSA contains more than one ABR, OSPF elects one of the ABRs to perform the LSA translation for NSSA. OSPF elects the ABR with the highest router ID. If the elected ABR becomes unavailable, OSPF automatically elects the ABR with the next highest router ID to take over translation of LSAs for the NSSA. The election process for NSSA ABRs is automatic.

OSPF Distribute List

This feature on Brocade NetIron products configures a distribution list to explicitly deny specific routes from being eligible for installation in the IP route table. By default, all OSPF routes in the OSPF route table are eligible for installation in the IP route table.

This feature does not block receipt of LSAs for the denied routes. The Layer 3 Switch still receives the routes and installs them in the OSPF database. The feature only prevents the software from installing the denied OSPF routes into the IP route table.

Intermediate System-to-Intermediate System (IS-IS) Concepts

The Intermediate System-to-Intermediate System (IS-IS) protocol is a link state Interior Gateway Protocol (IGP) that is based on the International Standard for Organization/International Electrotechnical Commission (ISO/IEC) Open Systems Internet Networking model (OSI). In IS-IS, an intermediate system (router) is designated as either a Level 1 or Level 2 router. A Level 1 router routes traffic only within the area in which the router resides. A Level 2 router routes traffic between areas within a routing domain.

Each route source has a default administrative distance. The default administrative distance for IS-IS is 115.

Configuring ISIS Authentication at the Router ISIS Mode

To configure ISIS authentication at the Router ISIS mode on a Brocade device, you must perform the following tasks:

- Configure ISIS Authentication Mode
- Configure ISIS Authentication Key

Configuring ISIS Authentication Mode

The following commands configure the ISIS for the authentication mode.

```
Brocade(config)# router isis
Brocade(config-isis-router)# auth-mode md5 level-1
Syntax: [no] auth-mode [cleartext | md5] [level-1 | level-2]
```

The **cleartext** parameter specifies that the ISIS PDUs will be authenticated using a cleartext password.

The **md5** parameter specifies that the ISIS PDUs will be authenticated using the Hashed Message Authentication codes - Message Digest 5 (HMAC-MD5) algorithm.

The **level-1** parameter specifies that the authentication type-length-value (TLV) tuple be added to the L1 LSP, L1 CSNP, and L1 PSNP packets.

The **level-2** parameter specifies that the authentication TLV tuple be added to the L2 LSP, L2 CSNP, and L2 PSNP packets.

Configuring ISIS Authentication Key

The following commands configure the ISIS for the authentication key.

```
Brocade(config-isis-router)# auth-key supervisor level-1
Syntax: [no] auth-key <string> [level-1 | level-2]
```

The **<string>** variable specifies a text string that is used as an authentication password. The authentication mode must be configured before this value can be configured.

By default, the authentication key is encrypted. If you want the authentication key to be in clear text, insert a 0 between **auth-key** and **<string>**.

BGP Concepts

Best Path Selection Process

When multiple paths for the same route are known to a BGP4 router, the router uses the following algorithm to weigh the paths and determine the optimal path for the route.

The optimal path depends on various modifiable parameters:

- **Mandatory requirement**
Is the next hop accessible through an IGP route? If not, ignore the route.
- **Tie breakers when multiple paths exist**
 1. Use the path with the **largest weight**
 2. If the weights are the same, prefer the route with the **largest local preference**
 3. If the routes have the same local preference, prefer the route that was **originated locally** (by this BGP4 Layer 3 Switch)
 4. If the local preferences are the same, prefer the route with the **shortest AS-path**
 5. If AS-path lengths are the same, prefer route with the **lowest origin type**. From low to high, route origin types are valued as:
 - IGP is lowest
 - EGP is higher than IGP but lower than INCOMPLETE
 - INCOMPLETE is highest
 6. If the routes have the same origin type, prefer the route with the **lowest MED**



Note

The device compares the MEDs of two otherwise equivalent paths **if and only if** the routes were learned from the same neighboring AS. This behavior is called **deterministic MED**. Deterministic MED is always enabled and cannot be disabled.

7. Routes received through **EBGP** from a BGP4 neighbor
8. Routes received through an **IBGP** neighbor
9. If all the comparisons above are equal, prefer the route with the **lowest IGP metric** to the BGP4 next hop. This is the closest internal path inside the AS to reach the destination.
10. If the internal paths also are the same and BGP4 load sharing is enabled, load share among the paths. Otherwise, prefer the route that comes from the **BGP4 router with the lowest router ID**.

BGP Route Advertisements

By default a BGP router will advertise all BGP-learned routes to all BGP neighbors. This behavior can be modified using route-maps. Applying a route-map to a BGP neighbor statement allows you to specify the prefixes to be learned or advertised to a specific neighbor.

```
Router(config-bgp) # neighbor 1.2.3.4 route-map out ADV_INT_ONLY
```

The **route-map in | out <map-name>** specifies a route map the device will apply to updates sent to or received from the specified neighbor. The **in | out** keywords specify whether the list is applied on updates received from the neighbor or sent to the neighbor.



Note

The route map must already be configured.

Next-Hop-Self

The BGP command **neighbor x.x.x.x next-hop-self** may be applied to an IBGP neighbor. The **next-hop-self** argument specifies that the router should list itself as the next hop in updates sent to the specified neighbor, rather than letting the protocol choose the next hop. This option is disabled by default.

EBGP Multihop

EBGP speakers are usually directly connected (i.e. over a WAN link). Sometimes they cannot be directly connected. In this special case, the **neighbor x.x.x.x ebgp-multihop [<num>]** command is used. The **ebgp-multihop [<num>]** argument specifies that the neighbor is more than one hop away and that the session type with the neighbor is thus EBGP-multihop. This option is disabled by default. The **<num>** parameter specifies the TTL you are adding for the neighbor. You can specify a number from 0 – 255. The default is 0. If you leave the EBGP TTL value set to 0, the software uses the IP TTL value. Multihop is only used for EBGP, not for IBGP.

BGP Confederation

A confederation is a BGP4 Autonomous System (AS) that has been subdivided into multiple, smaller ASs. Subdividing an AS into smaller ASs simplifies administration and reduces BGP-related traffic, thus reducing the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP routers in the AS.

Normally, all BGP routers within an AS must be fully meshed, so that each BGP router has interfaces to all the other BGP routers within the AS. This is feasible in smaller ASs but becomes unmanageable in ASs containing many BGP routers.

When you configure BGP routers into a confederation, all the routers within a sub-AS (a subdivision of the AS) use IBGP and must be fully meshed. However, routers use EBGP to communicate between different sub-ASs.

To configure a confederation, configure groups of BGP routers into sub-ASs. A sub-AS is simply an AS. The term “sub-AS” distinguishes ASs within a confederation from ASs that are not in a confederation. For the viewpoint of remote ASs, the confederation ID is the AS ID. Remote ASs do not know that the AS represents multiple sub-ASs with unique AS IDs.

You can use any valid AS numbers for the sub-ASs. If your AS is connected to the Internet, Brocade recommends that you use numbers from within the private AS range (64512 – 65535). These are private ASs numbers and BGP4 routers do not propagate these AS numbers to the Internet.

Fast External Failover

BGP4 devices rely on KEEPALIVE and UPDATE messages from neighbors to signify that the neighbors are alive. For BGP4 neighbors that are two or more hops away, such messages are the only indication that the BGP4 protocol has concerning the alive state of the neighbors. As a result, if a neighbor becomes non-operational, the device waits until the Hold Time expires or the TCP connection fails before concluding that the neighbor is not operational and closing its BGP4 session and TCP connection with the neighbor.

The device waits for the Hold Time to expire before ending the connection to a directly-attached BGP4 neighbor that becomes non-operational.

For directly-attached neighbors, the device immediately senses loss of a connection to the neighbor from a change of state of the port or interface that connects the device to the neighbor. For directly-attached EBGP neighbors, the device uses this information to immediately close the BGP4 session and TCP connection to locally attached neighbors that become non-operational.



Note

The fast external failover feature applies only to directly attached EBGP neighbors. The feature does not apply to IBGP neighbors.

Customizing BGP4 Multipath Load Sharing

By default, when BGP4 Multipath Load Sharing is enabled, both IBGP and EBGP paths are eligible for load sharing, while paths from different neighboring ASs are not eligible. You can change load sharing to apply only to IBGP or EBGP paths, or to support load sharing among paths from different neighboring ASs.

To enable load sharing of IBGP paths only, enter the following command at the BGP4 configuration level of the CLI.

```
Brocade(config-bgp) # multipath ibgp
```

To enable load sharing of EBGP paths only, enter the following command at the BGP4 configuration level of the CLI.

```
Brocade(config-bgp) # multipath ebgp
```

To enable load sharing of paths from different neighboring ASs, enter the following command at the BGP4 configuration level of the CLI.

```
Brocade(config-bgp) # multipath multi-as
```

Syntax: `[no] multipath ebgp | ibgp | multi-as`

The `ebgp | ibgp | multi-as` parameter specifies the change you are making to load sharing:

- **ebgp** – Load sharing applies only to EBGP paths. Load sharing is disabled for IBGP paths.
- **ibgp** – Load sharing applies only to IBGP paths. Load sharing is disabled for EBGP paths.
- **multi-as** – Load sharing is enabled for paths from different ASs.

Router Reflector

A BGP router selects a preferred BGP4 route for a specific prefix learned from multiple peers by using the BGP best path selection algorithm, and installs the BGP4 route in the Routing Table Manager (RTM). The BGP router marks the preferred BGP4 route as the best route, and advertises the route to other BGP4 neighbors. Generally, the RTM route table size is larger than the number of unique BGP4 routes in the BGP4 route table. All preferred BGP4 routes are installed in RTM and are marked as the best BGP4 routes.

However, in certain configurations it is possible that the total number of preferred BGP4 routes may exceed the RTM route table size limit. Therefore, some preferred BGP4 routes may not be installed in the RTM, and the BGP router is not able to forward traffic correctly for those BGP4 routes. Those BGP4 routes are not considered as the best BGP4 routes, and are not advertised to other BGP4 neighbors because traffic mis-forwarding or packet drop can occur.

When a BGP router is configured as only a route reflector server, and is not placed directly in the forwarding path, it is possible to mark all preferred BGP4 routes as the best routes to be advertised to other BGP4 neighbors even if the routes are not installed in the RTM. To support the behavior of a BGP router as a route reflector server in such a scenario, use the **always-propagate** command and the **rib-route-limit** command.

General IP Routing Protocol Concepts

IP Route Selection and Administrative Distance

IP routers use a lookup mechanism. IP lookup is an important action in router that is to find the next hop of each incoming packet with a longest-prefix-match address in the routing table. In other words, when a router determines the path to a certain destination, it will first choose the entry with the longest prefix match.

There may be multiple entries learned from different sources for the same destination network and these entries have the same prefix length. These different sources may be BGP4, OSPF, RIP, static routes, and so on. The software compares the routes on the basis of each route's administrative distance. If the administrative distance of the paths is lower than the administrative distance of paths from other sources (such as static IP routes, RIP, or OSPF), the BGP4 paths are installed in the IP route table.

Table 6 lists the default administrative distances which are found on the Brocade router.

TABLE 6 Default Administrative Distances

Protocol	Cost
Directly connected	0 (this value is not configurable)
Static	1 (applies to all static routes, including default routes)
External BGP (eBGP)	20
OSPF	110
RIP	120
Internal BGP (iBGP)	200
Local BGP	200
Unknown	255 (the router will not use this route)

Lower administrative distances are preferred over higher distances. For example, if the router receives routes for the same network from OSPF and from RIP, the router will prefer the OSPF route by default.

Bidirectional Forwarding Protection (BFD)

Multi-Service IronWare software provides support for Bidirectional Forwarding Detection (BFD). BFD provides rapid detection of the failure of a forwarding path by checking that the next-hop device is alive. Without BFD enabled, it can take from 3 to 30 seconds to detect that a neighboring device is not operational causing packet loss due to incorrect routing information at a level unacceptable for real-time applications such as VOIP and video over IP. Using BFD, you can detect a forwarding path failure in 300 milliseconds or less depending on your configuration. The BFD Control Message is an UDP message with destination port 3784.

Policy-Based Routing (PBR)

Policy-Based Routing (PBR) allows you to use ACLs and route maps to selectively modify and route IP packets in hardware. The ACLs classify the traffic. Route maps that match on the ACLs set routing attributes for the traffic.

A PBR policy specifies the next hop for traffic that matches the policy. Using standard ACLs with PBR, you can route IP packets based on their source IP address. With extended ACLs, you can route IP packets based on all of the match criteria in the extended ACL.

You can configure the Brocade device to perform the following types of PBR based on a packet's Layer 3 and Layer 4 information:

- Select the next-hop gateway.
- Send the packet to the null interface (null0).

When a PBR policy has multiple next hops to a destination, PBR selects the first live next hop specified in the policy that is up. If none of the policy's direct routes or next hops is available, the packets are forwarded as per the routing table.

6 – MPLS Concepts

After reviewing this section be sure you can perform the following:

- Describe basic MPLS concepts
- Demonstrate knowledge of MPLS VPNs
- Demonstrate knowledge of MPLS configuration

How MPLS Works

MPLS uses a *label switching* forwarding method to direct packets through a network. In label switching, a packet is assigned a label and passes along a predetermined path of routers. Forwarding decisions are based on the contents of the label, rather than information in the packet's IP header.

The following sections describe these basic MPLS concepts:

- How packets are forwarded through an MPLS domain
- The kinds of Label Switched Paths (LSPs) that can be configured on a device
- The components of an MPLS label header

How Packets are Forwarded Through an MPLS Domain

An *MPLS domain* consists of a group of MPLS-enabled routers, called **LSRs** (Label Switching Routers). In an MPLS domain, packets are forwarded from one MPLS-enabled router to another along a predetermined path, called an **LSP** (Label Switched Path). LSPs are one-way paths between MPLS-enabled routers on a network. To provide two-way traffic, you configure LSPs in each direction.

The LSRs at the headend and tailend of an LSP are known as **LERs** (Label Edge Routers). The LER at the headend, where packets enter the LSP, is known as the **ingress LER**. The LER at the tailend, where packets exit the LSP, is known as the **egress LER**.

Each LSP has one ingress LER and one egress LER. The ingress LER is responsible for inserting (called pushing) a label, or stack of labels into the packet. The egress LER is responsible for removing (called popping) the label from the packet. Packets in an LSP flow in one direction: from the ingress LER towards the egress LER.

In between the ingress and egress LERs there may be zero or more **transit LSRs**. The transit LSR is responsible for swapping labels throughout the LSP. A device enabled for MPLS can perform the role of ingress LER, transit LSR, or egress LER in an LSP. Further, a device can serve simultaneously as an ingress LER for one LSP, transit LSR for another LSP, and egress LER for some other LSP.

LSRs may drop IP or labeled packets in the following scenarios:

- A labeled packet is received but the label is not in the Label Forwarding Table, even if the IP destination is in the IP routing table.
- An IP packet is received but the destination IP address is not in the IP routing table, even if there is an LSP to the destination.

[Figure 11](#) depicts an MPLS domain with a single LSP consisting of three LSRs: an ingress LER, a transit LSR, and an egress LER.

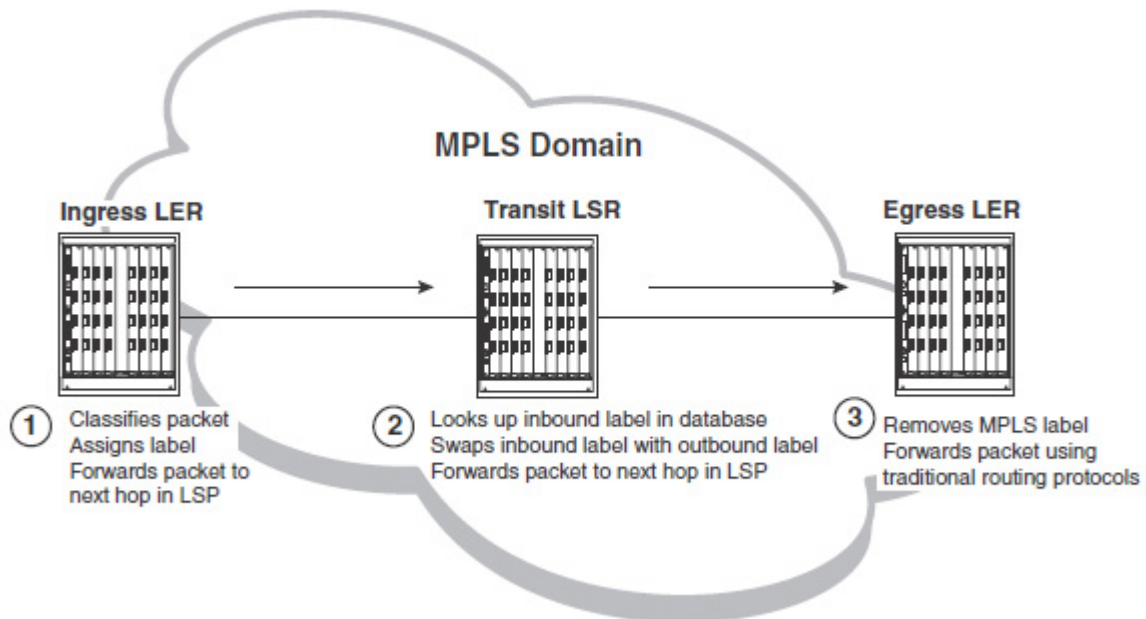


Figure 11: Label Switching in an MPLS Domain

Types of LSPs

An LSP in an MPLS domain can be either **static** or **signalled**.

Signalled LSPs

Signalled LSPs are configured at the ingress LER only. When the LSP is enabled, RSVP signalling messages travel to each LSR in the LSP, reserving resources and causing labels to be dynamically associated with interfaces. When a packet is assigned to a signalled LSP, it follows a pre-established path from the LSP's ingress LER to its egress LER. This path can be one of the following:

- A path that traverses an explicitly specified set of MPLS routers
- The IGP shortest path across the MPLS domain, determined from local routing tables
- A traffic-engineered path calculated by the device using constraints such as bandwidth reservations, administrative groups, and network topology information

Penultimate Hop Popping

On signalled LSPs, the MPLS label is popped at the next-to-last LSR in the LSP, instead of at the egress LER. This action is called *penultimate hop popping*. Penultimate hop popping improves forwarding efficiency by allowing the egress LER to avoid performing both a MPLS forwarding table lookup and an IP forwarding table lookup for each packet exiting the LSP. Instead, the MPLS label is popped at the penultimate (next-to-last) LSR, and the packet is forwarded to the egress LER with no MPLS encoding. The egress LER, in fact, does not recognize the packet as emerging from an LSP.

When an LSR receives an MPLS packet, it looks up the label in its MPLS forwarding table. Normally, this table maps the label and inbound interface to a new label and outbound interface. However, when this is the penultimate LSR in an LSP, the label and inbound interface map only to an outbound interface. The penultimate LSR pops the label and forwards the packet – now a regular IP packet – out the outbound interface. When the packet reaches the egress LER, there is no indication that it had been forwarded over an LSP. The packet is forwarded using standard hop-by-hop routing protocols.

MPLS Label Header Encoding

The following diagram illustrates the structure of the 32-bit MPLS label header. When a packet enters an LSP, the ingress LER pushes a label onto the packet. An MPLS label header is composed of the following parts:

Label value (20 bits)

The label value is an integer in the range 16 – 1048575. (Labels 0 – 15 are reserved by the IETF for special usage.) For signalled LSPs, the device dynamically assigns labels in the range 1024 – 499999.

EXP field (3 bits)

The EXP field is designated for experimental usage. By default, a device uses the EXP field to define a Class of Service (CoS) value for prioritizing packets travelling through an LSP.

S (Bottom of Stack) field (1 bit)

An MPLS packet can be assigned multiple labels. If an MPLS packet has multiple labels, they are logically organized in a last-in, first-out *label stack*. An LSR performs a pop or swap operation on the topmost label; that is, the most recently applied label in the stack. The Bottom of Stack field indicates whether this label is the last (oldest) label in the stack. If the label is the last one in the stack, the Bottom of Stack field is set to 1. If not, the Bottom of Stack field is set to 0.

A device acting as an LSR can perform one push, swap, or pop operation on an incoming MPLS packet. The device can accept MPLS packets that contain multiple labels, but only the topmost label is acted upon.

TTL field (8 bits)

The TTL field indicates the Time To Live (TTL) value for the MPLS packet. At the ingress LER, an IP packet's TTL value is copied to its MPLS TTL field. At each transit LSR hop, the MPLS TTL value is decremented by 1. If the MPLS TTL value reaches 0, the packet is discarded. Optionally, you can configure the LSRs not to decrement the MPLS TTL value at each hop.

Configuring Redundant Paths for an LSP

A signalled LSP has a primary path, which is either user-defined or computed by the ingress LER. Optionally, you can configure one or more redundant paths to serve as a backup. If the primary path fails, traffic for the LSP can be forwarded over the redundant path. When no redundant path is configured for the LSP, if the primary path fails, the ingress LER automatically attempts to compute a new path to the egress LER, establish the new path, and then redirect traffic from the failed path to the new path.

By default, the redundant path is established only when the primary path fails. You can optionally configure a redundant path to operate in **hot-standby** mode. A hot-standby path is established at the same time the primary path is established. Resources are allocated to the hot-standby path, although no packets for the LSP are sent over the hot-standby path until the primary path fails. When the primary path fails, the already-established hot-standby path immediately takes over from the primary path. Since the hot-standby path is already active, service outages that can arise from the process of signaling and establishing a new path are eliminated.

Label Distribution Protocol (LDP)

Brocade supports Label Distribution Protocol (LDP) for the configuration of non-traffic-engineered tunnel LSPs in an MPLS network. LDP uses UDP and TCP port 646 for communication between peers. UDP is used for peer discovery and TCP is used to setup and maintain connection-oriented links between discovered peers. LDP is described in RFC 3036.

When used to create tunnel LSPs, LDP allows a set of destination IP prefixes (known as a Forwarding Equivalence Class or FEC) to be associated with an LSP. Each LSR establishes a peer relationship with its neighboring LDP-enabled routers and exchanges label mapping information. This label mapping information is stored in an LDP database on each LSR. When an LSR determines that one of its peers is the next hop for a FEC, it uses the label mapping information from the peer to set up an LSP that is associated with the FEC. It then sends label mapping information to its upstream peers, allowing the LSP to extend across the MPLS network.

The devices advertise their loopback addresses to their LDP peers as a 32-bit prefix-type FEC. When an LSR installs a label for a FEC, it also creates an MPLS tunnel route, which is then made available to routing applications. This allows each router to potentially be an ingress LER for an LSP whose destination is the device's loopback address.

The result of an LDP configuration is a full mesh of LSPs in an MPLS network, with each LDP-enabled router a potential ingress, transit, or egress LSR, depending on the destination.

LDP over RSVP (for Transit LSR only)

LDP over RSVP (for transit LSR only) enables LDP traffic to tunnel across RSVP tunnels. The RSVP tunnel is the transit of the LDP tunnel. On Brocade NetIron XMR and Brocade MLX series devices, LDP over RSVP can run over all types of LSPs (for example, one-to-one or facility Fast ReRoute (FRR) LSPs, adaptive LSPs, or redundant LSPs).

MPLS Layer 2 VPNs

MPLS Layer 2 VPNs provide LAN-type services over a MPLS network infrastructure. The goal of Layer 2 VPN solutions is to make the MPLS network appear as point-to-point (VLL) or point-to-multipoint (VPLS) links between two or more customer sites. Because this is a Layer 2 solution, customer IP routes are not learned and customer MAC addresses are preserved across the MPLS network.

Virtual Leased Lines (VLL)

Virtual Leased Line is also known as Pseudo Wire Emulation as defined by the IETF PWE3 Working Group. MPLS VLL is a method for providing point-to-point Ethernet or VLAN connectivity over an MPLS domain.

Specifying a VLL Peer

The VLL peer is the PE router at the other end of the VLL. As part of VLL configuration, you specify the loopback IP address of the VLL peer.

Each PE router must have tunnel LSP reachability to its VLL peer. Tunnel LSP reachability is defined as having at least one operational LSP tunnel with the destination (the LSP's "to" address) matching the VLL peer's IP address. An LSP terminating on the VLL peer but configured with a different destination address would not be considered a match.

By default, each PE router attempts to initiate an LDP session through extended discovery with its VLL peer, if a session is not already established. The PE router also allocates a VC label from a per-platform label range that is mapped to the local endpoint. Once the LDP session is established, the locally assigned VC label, along with the VLL VC ID is advertised to the VLL peer in a downstream-unsolicited manner. In a similar way, the PE also learns the remotely assigned VC label from the VLL peer.

Specifying a VLL Endpoint

The endpoint of a VLL specifies what happens to packets exiting the VLL. You set the endpoint on the local PE router and this endpoint is mapped to a VC label. The VC label is advertised to the remote PE router at the other end of the VLL through LDP. The remote PE router applies this label to packets entering the VLL. When the packet reaches the end of the VLL through the MPLS uplink, the local PE router checks the mapping between the VC label and the endpoint, removes the VC label from the packet, and forwards the packet out the port specified as the endpoint.

All VLL endpoints can be dual-mode ports (tagged-untagged). An untagged endpoint port is removed from the default VLAN and cannot be added back to the default VLAN. A VLL endpoint can be tagged in multiple VLL and L2 VLANs and untagged in one other VLAN.

The Customer Edge (CE) device is connected to the PE router over an untagged, dual-tagged, or single-tagged port.

- With a *single-tagged* port, each pair (port, VLAN ID) is identified as a unique endpoint, and the packets are sent in tagged Ethernet format.
- In the case of an *untagged* port, an endpoint is identified by the physical port alone, and the packets are sent in untagged Ethernet format.
- In the case of a *dual-tagged* port, the packets contain both an outer VLAN tag and an inner VLAN tag.

Special Considerations for VLL Dual-tagged Endpoints

To change an existing single-tagged VLL endpoint to a dual-tagged endpoint, first delete the VLAN configuration, then configure the endpoint as dual-tagged.

Specifying a LAG Group as the Endpoint of a VLL

The endpoint of a VLL can be a LAG group. When the endpoint of a VLL is a LAG group, the VLL traffic load is distributed to the customer edge (CE) device across all of the LAG group's ports, using a hashing mechanism.

If you first create a LAG and then configure a VLL instance, the port you specify as the VLL endpoint must also be the port you specified as the primary port of the LAG group:

- If you later delete the LAG from the configuration, only the primary port will still be a port of the VLL and all secondary ports will become normal ports.
- If you specified a tagged endpoint for the VLL instance, all of the ports in the LAG must be tagged.
- Traffic received from any port in the LAG is forwarded to the VLL instance. All traffic is matched to its VLAN.
- Both static and dynamic LAGs are supported.

Virtual Private LAN Services (VPLS)

Virtual Private LAN Services (VPLS) enhances the point-to-point connectivity defined in the Draft-Martini IETF documents by specifying a method for virtual circuits (VCs) to provide point-to-multipoint connectivity across the MPLS domain, allowing traffic to flow between remotely connected sites as if the sites were connected by a Layer 2 switch.

Specifying VPLS Peers

As part of the VPLS configuration, you specify the IP address of each VPLS peer. VPLS requires a full mesh of tunnel LSPs; each PE router must have tunnel LSP reachability to each of its VPLS peers. Tunnel LSP reachability is defined as having at least one operational RSVP- or LDP-signalled LSP with the destination (the "to" address of the LSP) matching the VPLS peer's IP address. An LSP terminating on the VPLS peer but configured with a different destination address would not be considered a match.

By default, each PE router attempts to initiate an LDP session through extended discovery with its VPLS peers, if a session is not already established. Each VPLS instance is allocated a range of 32 labels. The PE router assigns one label in the range to each of its peers to be used as the peer's local VC label. If there are more than 32 peers in the VPLS instance, an additional label range is automatically allocated to the VPLS instance. The size of the label range depends on the configured maximum number of VPLS instances on the device.

BGP4 must be enabled on the device and a local Autonomous System (AS) number must be assigned before VPLS auto-discovery can be enabled.

Setting a per-VPLS MAC Table Limit

You can configure a maximum number of MAC entries that can be learned for a specified VPLS instance. This number cannot be exceeded. This limit can be configured at any time, although operation is more robust if you configure the limit at the same time that you configure the VPLS instance.

Flooding Layer 2 BPDUs in VPLS

By default, Layer 2 STP and Per VLAN Spanning Tree (PVST) Bridge Protocol Data Units (BPDUs) entering a VPLS endpoint are not transparently flooded within the VPLS instance. The BPDUs are dropped when they enter the VPLS endpoint. This behavior can be changed on a per-physical-port basis. Because the BPDU block option is configurable per physical interface, it will affect all VPLS instances that have endpoints on that interface.

To flood BPDUs in VPLS, use the following command.

```
Brocade(config)# interface ethernet 1/1  
Brocade(config-if-e10000-1/1)# no vpls-bpdu-block  
Syntax: [no] vpls-bpdu-block
```

VPLS Tagged Mode

VPLS tagged mode enables the preservation of the VLAN tag information in the payload. In VPLS tagged mode, the VLAN priority of the original (incoming) packets is carried across the MPLS cloud to remote peers.

By default, VPLS packets are sent across the MPLS cloud in raw mode.

MPLS Layer 3 VPNs

MPLS Layer 3 VPNs provide customer routing services over a MPLS network infrastructure. With Layer 3 VPNs the provider routers participate in customer routing, ensuring optimal routing between sites over the MPLS network. Providers are able to carry separate routes for each customer it supports over the MPLS network, even if those customers use overlapping IP addresses.

Multi-VRF

Multi-VRF provides the ability to maintain multiple “Virtual Routing and Forwarding” (VRF) tables on the same Provider Edge (PE) Router. Multi-VRF uses multiple instances of a routing protocol such as BGP or OSPF to exchange route information for a VPN among peer PE routers. The Multi-VRF capable PE router maps an input customer interface to a unique VPN instance. The router maintains a different VRF table for each VPN instance on that PE router. Multiple input interfaces may also be associated with the same VRF on the router, if they connect to sites belonging to the same VPN. This input interface can be a physical interface or a virtual Ethernet interface on a port.

Multi-VRF routers communicate with one another by exchanging route information in the VRF table with the neighboring PE router. This exchange of information among the PE routers is done using BGP or OSPF. The PE routers that communicate with one another should be directly connected at Layer 3. Customers connect to PE routers in the network using Customer Edge (CE) routers as shown in [Figure 12](#).

[Figure 12](#) depicts a network using Multi-VRF to provide connectivity among sites that belong to multiple VPNs. To share the VPN route table information with remote PEs, each PE creates separate virtual interfaces and run different instances of the PE-PE routing protocol for each VRF.

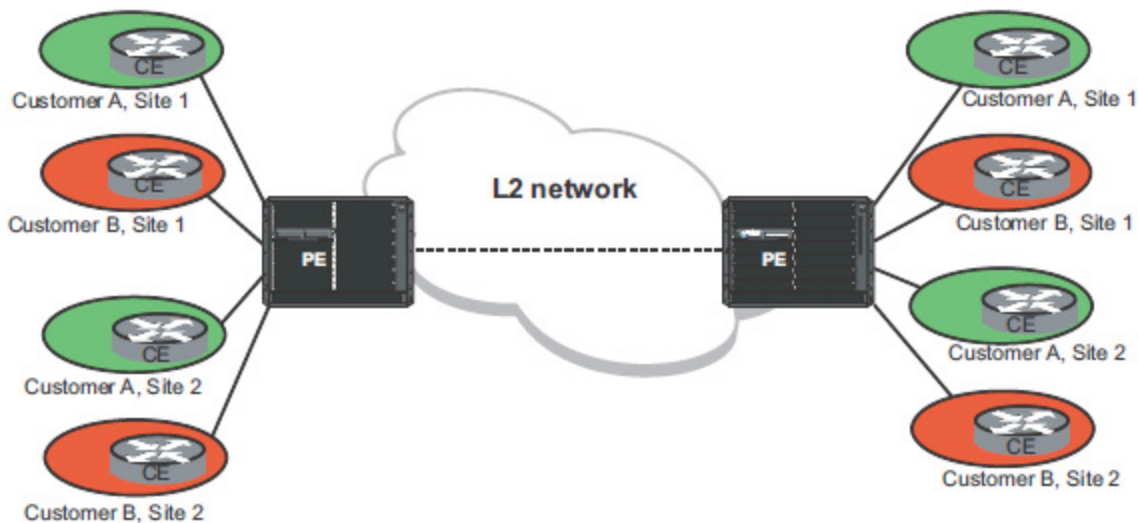


Figure 12: Multi-VRF Network

Multi-VRF and BGP or MPLS VPNs share some common aspects. For instance, in both cases the edge router maintains a VRF for all directly connected sites that are part of the same VPN. Also in both cases, the PE and CE routers share customer route information using a variety of PE-CE routing protocols, such as OSPF, RIP, E-BGP or static routes. Overlapping address spaces among different VPNs are allowed for both.

There are however, several differences between the two VPN technologies. The fundamental difference between the two technologies is that Multi-VRF requires that peering PE routers be directly connected at Layer 3. A Layer 2 network however, can be present between these directly-connected PE routers. BGP or MPLS VPNs do not have this restriction. In BGP or MPLS VPNs, the MPLS network determines the path to the peer router. In order to distinguish between devices with overlapping IP addresses, route targets are used in BGP or MPLS VPNs. Multi-VRF uses the input interface to uniquely identify the associated VPN, which is why the two PE routers should be directly connected at Layer 3.

Benefits and Applications of Multi-VRF

Multi-VRF provides a reliable mechanism for a network administrator to maintain multiple virtual routers on the same device. The goal of providing isolation among different VPN instances is accomplished without the overhead of heavyweight protocols used in secure VPN technologies or the administrative complexity of MPLS VPNs. It is particularly effective when operational staff has expertise in managing IP networks but may not have the same familiarity in managing MPLS networks. Overlapping address spaces can be maintained among the different VPN instances.

Configuring a VRF

Follow the steps below to configure a VRF:

1. Change to the interface configuration context

```
Router(config)# interface ethernet 3/1
```

2. Enable VRF forwarding on the interface

```
Router(config-if-3/1)# vrf forwarding VRF-A
```



Note

Executing the **vrf forwarding** command will remove any existing IP configuration from the interface

3. Configure an IP address on the interface

```
Router(config-if-3/1)# ip address 1.2.3.1/24
```

BGP Extended Attributes for Layer 3 VPNs

Assigning a Route Distinguisher to a VRF

Each instance of a VRF must have a unique Route Distinguisher (RD) assigned to it. The RD is prepended on any address being routed or advertised. The RD can be defined as either ASN-relative or IP address-relative. Since the RD is unique to an instance of a VRF, it allows the same IP address to be used in different VPNs without creating any conflict.

To assign a Route Distinguisher (RD) for a VRF based on the AS number 3 and the arbitrary identification number 6, enter the following command.

```
Brocade(config-vrf)# rd 3:6
```

Syntax: [no] **rd** <route_distinguisher>

Defining Automatic Route Filtering

Each VRF is configured with import and export route targets. The export route target sets an extended community attribute number that is appended to all routes that are exported from the VRF. The import route target value sets a filter that determines the routes that will be accepted into the VRF. Any route with a value in its import route-target contained in its extended attributes field matching the value in the VRF's import route target will be accepted. Otherwise the route will be rejected. This process is referred to as automatic route filtering.

To define an import route target of 3:6 and an export route target of 3:8 for a VPN, enter the following commands.

```
Brocade(config-vrf)# route-target import 3:6
```

```
Brocade(config-vrf)# route-target export 3:8
```

Syntax: [no] **route-target** [import | export | both] <route-target>

This command associates a route target specified by the route-target variable with a specified VRF for control on routes.

The **import** parameter specifies that routes with route-target extended community attributes matching the specified route-target variable can be imported into the VRF where this command is configured.

The **export** parameter specifies the route-target extended community attributes that are attached to routes export from the specified VRF.

The **both** parameter specifies that both the import and export values apply to the specified route-target variable for the VRF where this command is configured. This is the default state. It applies if no specific value for this parameter is set.

The <route-target> variable specifies a target VRF extended community. Like a route distinguisher, it is either AS-relative or IP address-relative.

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. In version 03.5.00, the LAG index calculation for MPLS LAGs always included the IP and MAC source and destination addresses in addition to the MPLS label, such that the default behavior for hashing is based on the MPLS labels, including both IP and MAC source and destination addresses. In version 03.6.00, the following additional CLI commands can be added to restrict the hashing to just **mpls-ip** and **mpls-enet** respectively.

Using IP Source and Destination Addresses for Load Sharing

You can use the load-balance speculate-mpls-ip command to include the IP source and destination addresses in the calculation of the LAG index for a traffic flow within MPLS LAGs, as shown in the following.

```
Brocade(config)# load-balance speculate-mpls-ip all
```

Syntax: [no] **load-balance speculate-mpls-ip** [all | <slot-number> | <slot-number> <np-id>]



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.

```
Brocade(config)# load-balance speculate-mpls-enet all
```

Syntax: `[no] load-balance speculate-mpls-enet [all | <slot-number> | <slot-number> <np-id>]`



Note

The `load-balance speculate-mpls-enet` command will hash only on the Ethernet header portion.

7 – IP Multicast

After reviewing this section be sure you can perform the following:

- Demonstrate knowledge of multicast on a service provider network

Protocol Independent Multicast (PIM)

PIM was introduced to simplify some of the complexity of the routing protocol at the cost of additional overhead tied with a greater replication of forwarded multicast packets. PIM is similar to DVMRP in that PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two modes in which PIM operates: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

PIM primarily differs from DVMRP by using the IP routing table instead of maintaining its own, thereby being routing protocol independent.

PIM Dense

The Brocade device supports PIM DM V1 and V2. The default is V2. You can specify the version on an individual interface basis.

The primary difference between PIM DM V1 and V2 is the methods the protocols use for messaging:

- PIM DM V1 – uses the IGMP to send messages.
- PIM DM V2 – sends messages to the multicast address 224.0.0.13 (ALL-PIM-ROUTERS) with protocol number 103.

Dense mode, as the name implies is a dense, or flood and prune, multicast protocol. It implicitly builds shortest-path trees by flooding multicast traffic domain wide, and then pruning back branches of the tree where no receivers are present. PIM-DM is fairly easy to implement but generally has poor scaling ability.

PIM Sparse

Brocade devices support Protocol Independent Multicast (PIM) Sparse version 2. PIM Sparse provides multicasting that is especially suitable for widely distributed multicast environments. The Brocade implementation is based on RFC 2362.

In a PIM Sparse network, multicast traffic will not immediately be flooded by the Designated Router (DR). A PIM Sparse device that is connected to a host that wants to receive information for a multicast group must explicitly send a join request on behalf of the receiver (host) before traffic is flooded into the PIM domain.

Changing the Shortest Path Tree (SPT) Threshold

In a typical PIM Sparse domain, there may be two or more paths from a DR (designated router) for a multicast source to a PIM group receiver:

- **Path through the RP** – This is the path the device uses the first time it receives traffic for a PIM group. However, the path through the RP may not be the shortest path from the device to the receiver.

- **Shortest Path** – Each PIM Sparse device that is a DR for a multicast source calculates a shortest path tree (SPT) to all the PIM Sparse group receivers within the domain, with the device itself as the root of the tree. The first time a device configured as a PIM router receives a packet for a PIM receiver, the device sends the packet to the RP for the group. The device also calculates the SPT from itself to the receiver. The next time the device receives a PIM Sparse packet for the receiver, the device sends the packet toward the receiver using the shortest route, which may not pass through the RP.

By default, the device switches from the RP to the SPT after receiving the first packet for a given PIM Sparse group. The device maintains a separate counter for each PIM Sparse source-group pair.

After the device receives a packet for a given source-group pair, it starts a PIM data timer for that source-group pair. If the device does not receive another packet for the source-group pair before the timer expires, it reverts to using the RP for the next packet received for the source-group pair. In accordance with the PIM Sparse RFC recommendation, the timer is 210 seconds and is not configurable. The counter is reset to zero each time the device receives a packet for the source-group pair.

You can change the number of packets that the device sends using the RP before switching to using the SPT by entering commands such as the following.

```
Brocade(config)# router pim
Brocade(config-pim-router)# spt-threshold 1000
Syntax: [no] spt-threshold infinity | <num>
```

The **infinity** | <num> parameter specifies the number of packets. If you specify **infinity**, the device sends packets using the RP indefinitely and does not switch over to the SPT. If you enter a specific number of packets, the device does not switch over to using the SPT until it has sent the number of packets you specify using the RP.

Concurrent Support for Multicast Routing and Snooping

Multicast routing and multicast snooping instances work concurrently on the same device. For example, you can configure PIM routing on certain VEs interfaces and snooping on other VEs or VLANs. The limitation is that either multicast snooping or routing can be enabled on a VE interface or VLAN, but not on both. This is because all of the multicast data and control packets (IGMP, PIM) received on the snooping VLAN are handled by multicast snooping and do not reach the multicast routing component. Similarly, any multicast data or control packets received on a VE interface enabled with PIM or DVMRP routing are handled by the PIM or DVMRP routing component and are not seen by the IGMP or PIM snooping component.

The following considerations apply when configuring concurrent operation of Multicast Routing and Snooping.

1. Either multicast snooping or routing can be enabled on a VE or VLAN but not both.
2. Snooping can be enabled globally (`ip multicast <active | passive>`) as can multicast routing (`ip multicast-routing`).
3. The global snooping configuration is inherited by all current VLANs that are not enabled for multicast routing.
4. The global snooping configuration is also inherited by all new VLANs. Enabling multicast routing on a newly created VLAN or VE automatically disables snooping on the VLAN or VE.
5. When a VLAN-level snooping is configured, it is displayed.

Multicast Non-stop Routing

Multicast non-stop routing (NSR) provides hitless upgrade and switchover support for all IPv4 multicast, including default and non-default VRFs for IPv4 PIM-DM, PIM-SM, and PIM-SSM. Multicast NSR is not supported for IPv6 multicast and DVMRP. The software multicast state is kept in sync between the active and standby MPs. As the Brocade system enters a hitless upgrade or switchover state, the standby MP will take over as the new active MP. The new active MP will carry a pre-installed multicast state that was originally supported by the previous MP. The new active MP will revalidate the pre-installed multicast state, and pick up any new changes as needed before marking the multicast state as operational. When the LP is ready to complete the hitless upgrade or switchover process, the operational multicast state will be downloaded to the LP CPU. When the LP resets, and the outage of the LP CPU occurs, pre-existing hardware forwarding multicast traffic will continue to flow without disruption, and the hardware multicast forwarding state is retained in the LP hardware.

Multicast NSR is globally enabled across all VRFs by configuring the `ip multicast-nonstop-routing` command.

Multi-protocol Border Gateway Protocol (MBGP)

MBGP is an extension to BGP that allows a router to support separate unicast and multicast topologies. BGP4 cannot support a multicast network topology that differs from the network's unicast topology. MBGP allows you to support a multicast topology that is distinct from the network's unicast topology.

Configuring MBGP

1. Optional – Set the maximum number of multicast routes supported by the Brocade device
2. Enable MBGP by doing the following:
 - Enable PIM Sparse Mode (PIM SM) or PIM Dense Mode (PIM DM) globally and on the individual Reverse Path Forwarding (RPF) interfaces. PIM must be running on the Brocade device in order for the device to send multicast prefixes to other multicast devices.
 - Enable BGP4
3. Identify the neighboring MBGP routers
4. Optional – Configure an MBGP default route
5. Optional – Configure an IP multicast static route
6. Optional – Configure an MBGP aggregate address
7. Optional – Configure a route map to apply routing policy to multicast routes

8 – IPv6

After reviewing this section be sure you can perform the following:

- Demonstrate knowledge of IPv6 addressing
- Demonstrate knowledge of IPv6 routing

IPv6

An IPv6 address is 128 bits and is composed of 8 fields of 16-bit hexadecimal values separated by colons (:):

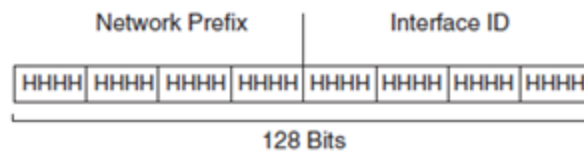


Figure 13: IPv6 Address Format

HHHH is a 16-bit hexadecimal value (0000 to FFFF), while H is a 4-bit hexadecimal value. The following is an example of an IPv6 address:

```
2001:0000:0000:0200:002D:D0FF:FE48:4672
```

Note that the IPv6 address includes hexadecimal fields of zeros. To make the address less cumbersome, you can do the following:

- Omit the leading zeros; for example, 2001:0:0:200:2D:D0FF:FE48:4672.
- Compress the successive groups of zeros at the beginning, middle, or end of an IPv6 address to two colons (::) once per address; for example, 2001::200:2D:D0FF:FE48:4672.
- When specifying an IPv6 address in a command syntax, keep the following in mind: You can use the two colons (::) only once in the address to represent the longest successive hexadecimal fields of zeros.
- The hexadecimal letters in IPv6 addresses are not case-sensitive.

Types of IPv6 Addresses

- **Unicast:** An address for a single interface. A packet sent to a unicast address is delivered to the interface identified by the address. There are several types of unicast addresses: Aggregatable global address (prefix 2000::/3), Local IPv6 unicast address (prefix FC00::/7), Link-local address (prefix FE80::/10), IPv4-compatible address (0:0:0:0:0:0:A.B.C.D), Loopback address (0:0:0:0:0:0:0:1 or ::1), and Unspecified address (0:0:0:0:0:0:0:0 or ::).



Note

The Local IPv6 unicast address (prefix FC00::/7), defined in RFC 4193, is the replacement for the Site-local address (prefix FEC0::/10), defined in RFC 3879, which has been deprecated.

This address range is a logical equivalent to the RFC 1918 (or private) addresses in IPv4.

- **Multicast:** An address for a set of interfaces belonging to different nodes. Sending a packet to a multicast address results in the delivery of the packet to all interfaces in the set. A multicast address has a fixed prefix of FF00::/8 (1111 1111). The next 4 bits define the address as a permanent or temporary address. The next 4 bits define the scope of the address (node, link, site, organization, global).
- **Anycast:** An address for a set of interfaces belonging to different nodes. Sending a packet to an anycast address results in the delivery of the packet to the closest interface identified by the address.

IPv6 Multicast Addresses

Routers and routing protocols that rely on multicast in IPv6 environments have assigned multicast addresses for their specific purpose. For example:

- FF02::1/64 – All hosts address
- FF02::2/64 – All routers address
- FF02::5/64 – OSPFv3 AllRouters
- FF02::6/64 – OSPFv3 AllDesignatedRouters
- FF02::9/64 – RIPng

Router ID in IPv6-only Networks

The router ID on Brocade devices is a 32-bit value denoted in an IPv4 address format. 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 9.9.9.9/24:
 - Loopback interface 1, 9.9.9.9/24
 - Loopback interface 2, 4.4.4.4/24
 - Loopback interface 3, 1.1.1.1/24
- If the device does not have any loopback interfaces, the default router ID is the lowest numbered IP interface configured on the device.
- Lastly, a router can be manually configured with the global **ip router-id** command.



Note

Because the router ID is a 32-bit value and an IPv6 address is 128 bits, a Brocade device with no IPv4 addresses configured cannot automatically assign the router ID. The router ID can only be assigned by executing the **ip router-id** command.

IPv6 Over IPv4 Tunnels

To enable communication between the isolated IPv6 domains using the IPv4 infrastructure, you can configure IPv6 over IPv4 tunnels.

Manual IPv6 Tunnel Configuration Notes

You can use a manually configured tunnel to connect two isolated IPv6 domains. You should deploy this point-to-point tunnel mechanism if you need a permanent and stable connection.

Configuration notes on manual tunnels:

- The tunnel mode should be **ipv6ip** indicating that this is ipv6 manual tunnel.
- Both source and destination addresses needs to be configured on the tunnel.
- On the remote side you need to have exactly opposite source/destination pair.
- The tunnel destination should be reachable through the IPv4 backbone.
- The ipv6 address on the tunnel needs to be configured for the tunnel to come up.
- The tunnel source can be an IP address or interface name.
- Manual tunnels provide static point-point connectivity.
- Static routing on top of the tunnel is supported.
- IPv6 routing protocols including OSPFv3 and RIPng on top of the tunnel are supported.

Configuring a Static IPv6 Route

You can configure a static IPv6 route to be redistributed into a routing protocol, but you cannot redistribute routes learned by a routing protocol into the static IPv6 routing table.

To configure a static IPv6 route for a destination network with the prefix `8eff::0/32` and a next-hop gateway with the link-local address `fe80::1` that the Brocade device can access through Ethernet interface `3/1`, enter the following command.

```
Brocade(config)# ipv6 route 8eff::0/32 ethernet 1 fe80::1
```

Syntax: `[no] ipv6 route <dest-ipv6-prefix>/<prefix-length> <next-hop-ipv6-address> [<metric>] [distance <number>]`

If you specify a link-local address, you must also specify the interface through which to access the address.

You can specify one of the following interfaces:

- An Ethernet interface
- A tunnel interface
- A virtual interface (VE)

9 – Monitoring, Maintenance, and Troubleshooting

After reviewing this section be sure you can perform the following:

- Demonstrate knowledge of troubleshooting techniques

Showing System Software

Most boot issues occur because incorrect or incompatible images have been downloaded. The **show version** command displays all versions that are currently running on the Brocade device.

```
Brocade# show version
```

Syntax: show version

Showing CPU Statistics

The first step in determining how your device is using memory and CPU is to get a view of the activity. Several **show** commands display information about CPU usage and CPU task activity.

```
Brocade# show tasks
```

Syntax: show tasks

```
Brocade# show cpu
```

Syntax: show cpu

Displaying Information for an Interface

To display information for a show interface for an ethernet port, enter the **show interface** command at any CLI level. The **show interface** command provides information about the configuration of the interface including, but not limited to, speed/duplex, STP properties, interface type, interface MAC address, interface input/output counters and errors.

```
Brocade# show interface ethernet 9/1
```

```
GigabitEthernet2/3 is up, line protocol is up
```

```
STP Root Guard is disabled, STP BPDU Guard is disabled
```

```
Hardware is GigabitEthernet, address is 0012.f298.4900 (bia 0012.f298.492a)
```

```
Configured speed auto, actual 1Gbit, configured duplex fdx, actual fdx
```

```
Member of VLAN 1 (untagged), 5 L2 VLANS (tagged), port is in dual mode (default vlan), port state is Forwarding
```

```
STP configured to ON, Priority is level0, flow control enabled
```

```
Priority force disabled, Drop precedence level 0, Drop precedence force disabled
```

```
arp-inspection-trust configured to OFF
```

```
mirror disabled, monitor disabled
```

```
Not member of any active trunks
```

```
Not member of any configured trunks
```

```
Port name is ->7.blade1.shelf1.access.aprd
```

```
MTU 1544 bytes, encapsulation ethernet
```

```
300 second input rate: 1509512 bits/sec, 713 packets/sec, 0.15% utilization
```

```
300 second output rate: 1992071 bits/sec, 751 packets/sec, 0.20% utilization
```

```
712896623 packets input, 204984611768 bytes, 0 no buffer
```

```
Received 1315502 broadcasts, 53313 multicasts, 711527808 unicasts
```

```
0 input errors, 0 CRC, 0 frame, 0 ignored
```

```
0 runts, 29433839 giants
```

```
NP received 712896745 packets, Sent to TM 712839428 packets
NP Ingress dropped 57317 packets
796106728 packets output, 366570033985 bytes, 0 underruns
Transmitted 2045784 broadcasts, 32330616 multicasts, 761730328 unicasts
0 output errors, 0 collisions
NP transmitted 796106833 packets, Received from TM 796534170 packets
```

Syntax: `show interface [ethernet <slot-port> [to <slot-port>]]`

You can display information for all ports in a device by using the `show interface` command without options, or use the `ethernet <slot-port>` option to limit the display to a single port, or add the `to <slot-port>` option for a range of ports.

To display information from the `show statistics` command for an Ethernet port, enter the following command at any CLI level. The `show statistics` command is used to display information about interface traffic, including but not limited to, in/out octets, in/out packets, Ethernet collisions, and errors.

```
Brocade# show statistics ethernet 9/1
PORT 9/1 Counters:
InOctets 210753498112 OutOctets 210753550720
InPkts 1646511726 OutPkts 1646512119
InBroadcastPkts 0 OutBroadcastPkts 0
InMulticastPkts 0 OutMulticastPkts 0
InUnicastPkts 1646511726 OutUnicastPkts 1646512142
InDiscards 0 OutDiscards 0
InErrors 0 OutErrors 0
InCollisions 0 OutCollisions 0
OutLateCollisions 0
Alignment 0 FCS 0
GiantPkts 0 ShortPkts 0
InBitsPerSec 3440829770 OutBitsPerSec 3440686411
InPktsPerSec 3360185 OutPktsPerSec 3360085
InUtilization 39.78% OutUtilization 39.78%
```

Syntax: `show statistics ethernet <slot/port>`

The `<slot/port>` variable specifies the port that you want to display statistics for.

LACP Trunking

The Link Aggregation Control Protocol (LACP) allows ports on both sides of a redundant link to automatically configure themselves into a trunk link (aggregate link), eliminating the need for manual configuration.

Brocade offers two types of link-aggregation groups (LAGs) that can utilize LACP:

- **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.
- **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.

Port Mirroring and Monitoring

Port mirroring is a method of monitoring network traffic that forwards a copy of each incoming or outgoing packet from one port on a network switch to another port where the packet can be analyzed. Port mirroring may be used as a diagnostic tool or debugging feature, especially for preventing attacks. Port mirroring can be managed locally or remotely.

Configure port mirroring by assigning a port from which to copy all packets, and a “mirror” port where the copies of the packets are sent (also known as the monitor port). A packet received on, or issued from, the first port is forwarded to the second port as well. Attach a protocol analyzer on the mirror port to monitor each segment separately. The analyzer captures and evaluates the data without affecting the client on the original port. The mirror port may be a port on the same switch with an attached RMON probe, a port on a different switch in the same hub, or the switch processor.

To configure port monitoring on an individual port on a Brocade device, enter commands similar to the following.

```
FastIron(config) #mirror-port ethernet 1/2/4
FastIron(config) #interface ethernet 1/2/11
FastIron(config-if-e1000-11) #monitor ethernet 1/2/4 both
```

Traffic on port e 1/2/11 will be monitored, and the monitored traffic will be copied to port e 1/2/4, the mirror port.

Syntax: [no] **mirror-port ethernet** [*<stack-unit>/<slotnum>/*]*<portnum>* [**input** | **output**]

Syntax: [no] **monitor ethernet** [*<stack-unit>/<slotnum>/*]*<portnum>* **both** | **in** | **out**

- The *<portnum>* parameter for mirror-port ethernet specifies the port to which the monitored traffic will be copied.
- The *<portnum>* parameter for monitor ethernet specifies the port on which traffic will be monitored.
- The **input** and **output** parameters configure the mirror port exclusively for ingress or egress traffic. If you do not specify one, both types of traffic apply.
- The **both**, **in**, and **out** parameters specify the traffic direction you want to monitor on the mirror port. There is no default.

To display the port monitoring configuration, enter the **show monitor** and **show mirror** commands. You may also configure ACL-based inbound mirroring, MAC filter-based mirroring, and VLAN-based mirroring.

Protecting Against Smurf Attacks

A *smurf attack* is a kind of DoS attack where an attacker causes a victim to be flooded with ICMP echo (ping) replies sent from another network. Figure 14 illustrates how a smurf attack works.

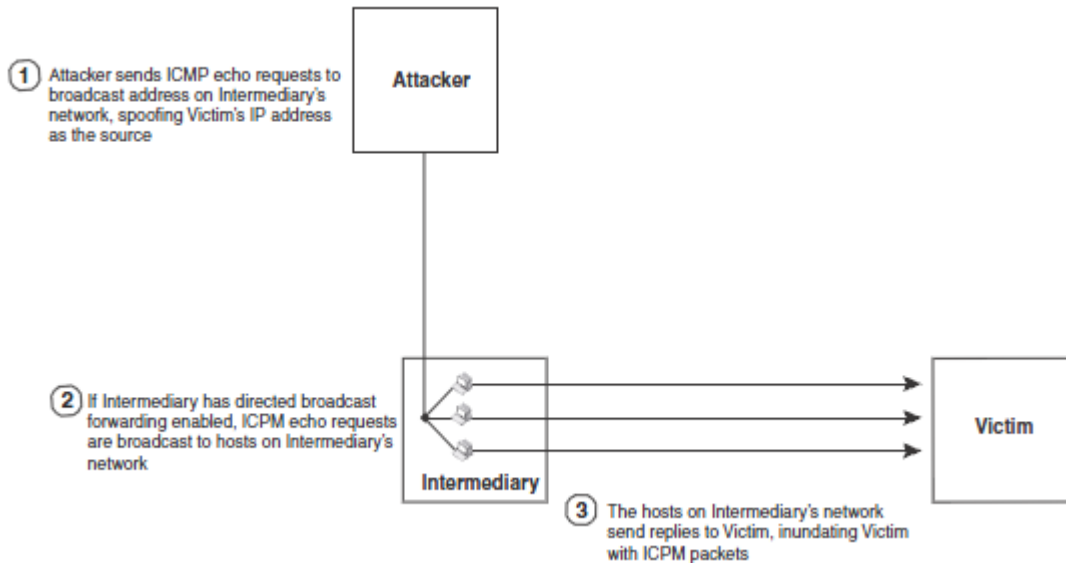


Figure 14: Smurf Attack

The attacker sends an ICMP echo request packet to the broadcast address of an intermediary network. The ICMP echo request packet contains the spoofed address of a victim network as its source. When the ICMP echo request reaches the intermediary network, it is converted to a Layer 2 broadcast and sent to the hosts on the intermediary network. The hosts on the intermediary network then send ICMP replies to the victim network.

For each ICMP echo request packet sent by the attacker, a number of ICMP replies equal to the number of hosts on the intermediary network are sent to the victim. If the attacker generates a large volume of ICMP echo request packets, and the intermediary network contains a large number of hosts, the victim can be overwhelmed with ICMP replies.

Avoid Being an Intermediary in a Smurf Attack

A smurf attack relies on the intermediary to broadcast ICMP echo request packets to hosts on a target subnet. When the ICMP echo request packet arrives at the target subnet, it is converted to a Layer 2 broadcast and sent to the connected hosts. This conversion takes place only when directed broadcast forwarding is enabled on the device.

To avoid being an intermediary in a smurf attack, make sure forwarding of directed broadcasts is disabled on the device. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, enter this command.

```
Brocade(config)# no ip directed-broadcast
```

Syntax: [no] ip directed-broadcast

BGP Neighbor States

The state of device sessions with each neighbor. The states are from this perspective of the device, not the neighbor. State values are based on the BGP4 state machine values described in RFC 1771 and can be one of the following for each device:

IDLE – The BGP4 process is waiting to be started. Usually, enabling BGP4 or establishing a neighbor session starts the BGP4 process

ADMND – The neighbor has been administratively shut down

CONNECT – BGP4 is waiting for the connection process for the TCP neighbor session to be completed

ACTIVE – BGP4 is waiting for a TCP connection from the neighbor



Note

If the state frequently changes between CONNECT and ACTIVE, there may be a problem with the TCP connection.

OPEN SENT – BGP4 is waiting for an Open message from the neighbor

OPEN CONFIRM – BGP4 has received an Open message from the neighbor and is now waiting for either a KEEPALIVE or NOTIFICATION message. If the device receives a KEEPALIVE message from the neighbor, the state changes to Established. If the message is a NOTIFICATION, the state changes to Idle

ESTABLISHED – BGP4 is ready to exchange UPDATE packets with the neighbor

Taking the Test

After the Introduction Screen, once you click on Next, you will see the following non-disclosure agreement:

IMPORTANT: PLEASE READ THE FOLLOWING BROCADE NON-DISCLOSURE CONFIDENTIALITY AGREEMENT CAREFULLY BEFORE TAKING THIS EXAM.

The following Non-Disclosure Confidentiality Agreement (the "Agreement") sets forth the terms and conditions of your use of the exam materials as defined below.

The Disclosure to you of this Exam and any questions, answers, worksheets, computations, drawings, diagrams, or any communications, including verbal communication by any party, regarding or related to the Exam and such Exam Materials and any derivatives thereof is subject to the Terms and Conditions of this Agreement.

You understand, acknowledge and agree:

- That the questions and answers of the Exam are the exclusive and confidential property of Brocade and are protected by Brocade intellectual property rights;
- That you may not disclose the Exam questions or answers or discuss any of the content of the Exam Materials with any person, without prior approval from Brocade;
- Not to copy or attempt to make copies (written, photocopied, or otherwise) of any Exam Material, including, without limitation, any Exam questions or answers;
- Not to sell, license, distribute, or give away the Exam Materials, questions, or answers;
- You have not purchased, solicited or used unauthorized (non-Brocade sanctioned) Exam Materials, questions, or answers in preparation for this exam;
- That your obligations under this Agreement shall continue in effect after the Exam and, if applicable, after termination of your credential, regardless of the reason or reasons for terminations, and whether such termination is voluntary or involuntary.

Brocade reserves the right to take all appropriate actions to remedy or prevent disclosure or misuse, including, without limitation, obtaining an immediate injunction. Brocade reserves the right to validate all results and take any appropriate actions as needed. Brocade also reserves the right to use any technologies and methods for verifying the identity of candidates. Such technology may include, without limitation, personally identifiable information, challenge questions, identification numbers, photographic information, and other measures to protect against fraud and abuse.

Neither this Agreement nor any right granted hereunder shall be assignable or otherwise transferable by you.

By clicking on the "A" button ("YES, I AGREE"), you are consenting to be bound by the terms and conditions of this agreement and state that you have read this agreement carefully and you understand and accept the obligations which it imposes without reservation. You further state that no promises or representations have been made to induce agreement and that you accept this agreement voluntarily and freely.

- A. YES, I AGREE
- B. NO, I DO NOT AGREE