BROCADE

CONFIGURATION GUIDE

Network OS Layer 2 Switching Configuration Guide, 7.0.1

Supporting Network OS 7.0.1

53-1004365-02
04 August 2016

Contents

Preface

Document conventions

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

Text formatting conventions

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

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

Command syntax conventions

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

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

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

Notes, cautions, and warnings

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

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

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

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

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

Brocade resources

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

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

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

Release notes are available on MyBrocade under Product Downloads.

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

Contacting Brocade Technical Support

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

Brocade customers

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

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

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

Brocade OEM customers

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

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

Document feedback

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

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

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

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

About this document

Supported hardware and software

In those instances in which procedures or parts of procedures documented here apply to some switches but not to others, this guide identifies exactly which switches are supported and which are not.

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

The following hardware platforms are supported by this release of Network OS:

- Brocade VDX 2741
- Brocade VDX 2746
- Brocade VDX 6740
  - Brocade VDX 6740-48
  - Brocade VDX 6740-64
- Brocade VDX 6740T
  - Brocade VDX 6740T-48
  - Brocade VDX 6740T-64
  - Brocade VDX 6740T-1G
- Brocade VDX 6940-36Q
- Brocade VDX 6940-144S
- Brocade VDX 8770
  - Brocade VDX 8770-4
  - Brocade VDX 8770-8

To obtain information about a Network OS version other than this release, refer to the documentation specific to that version.

Using the Network OS CLI

For complete instructions and support for using the Network OS command line interface (CLI), refer to the *Network OS Command Reference*.

What's new in this document

This document supports the following features introduced in Network OS7.0.1:

- VXLAN NSX Service Node load balancing
- MAC-move detection for VCS overlays
- Neighbor discovery for LACP on an RBridge

For complete information, refer to the *Network OS Release Notes*.

Edge-Loop Detection

Edge-loop detection overview

Edge-loop detection (ELD) detects and disables Layer 2 loops that would cause broadcast storms. Typically, these loops are caused by misconfigurations.

ELD is configured and enabled on Brocade VCS Fabric clusters. Any topology that includes one or more Brocade VCS Fabric clusters use ELD to detect Layer 2 loops and prevent broadcast storms. Standalone switches can be included in such a cluster, but loop detection takes place on the Brocade VCS Fabric cluster, and not on the standalone switch. You cannot use ELD in a network consisting of standalone switches only.

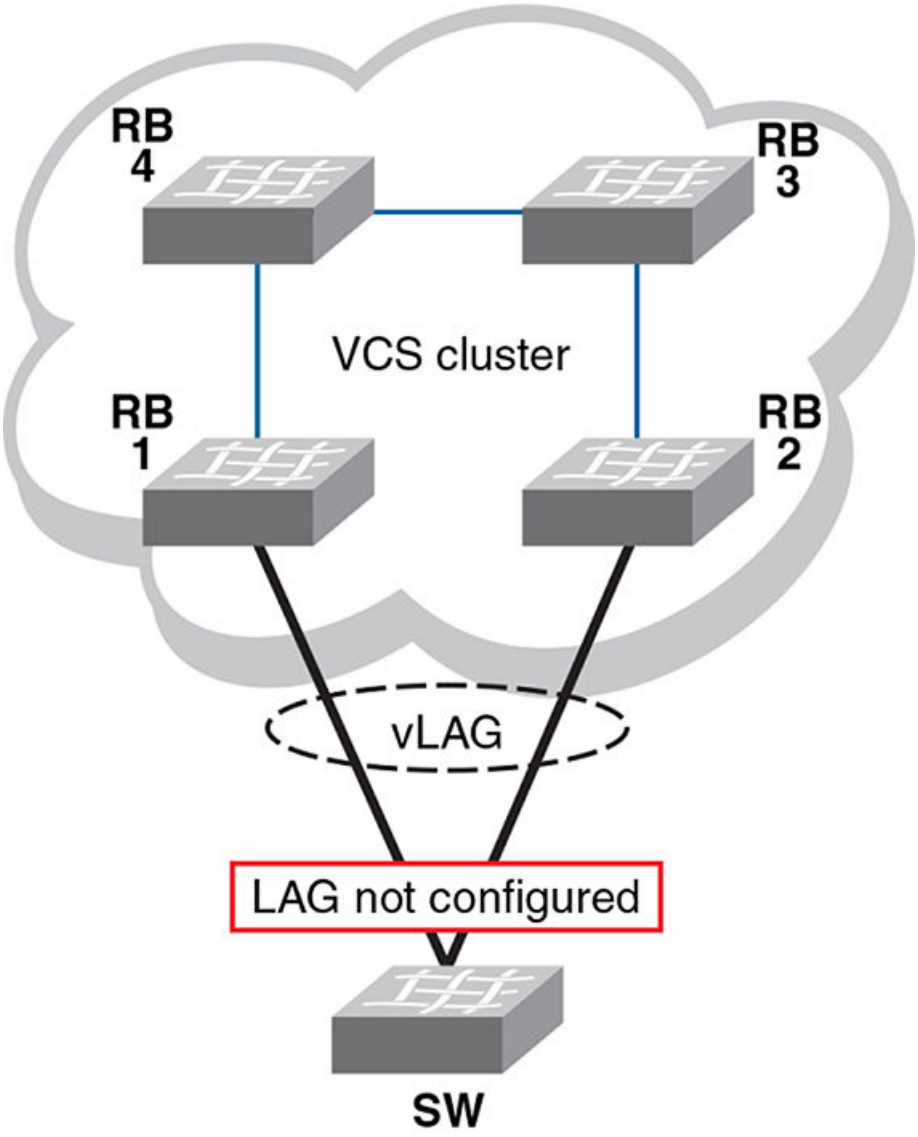
Specifically, ELD can be used to prevent broadcast storms caused by Layer 2 loops in the following topologies:

- A Brocade VCS Fabric cluster connects to a standalone switch.
- A Brocade VCS Fabric cluster connects to a multiple node network.
- A Brocade VCS Fabric cluster connects to other Brocade VCS Fabric clusters.

NOTE
For an additional method of detecting and resolving L2 loops, see Resolving Repeated MAC-Moves on page 305.

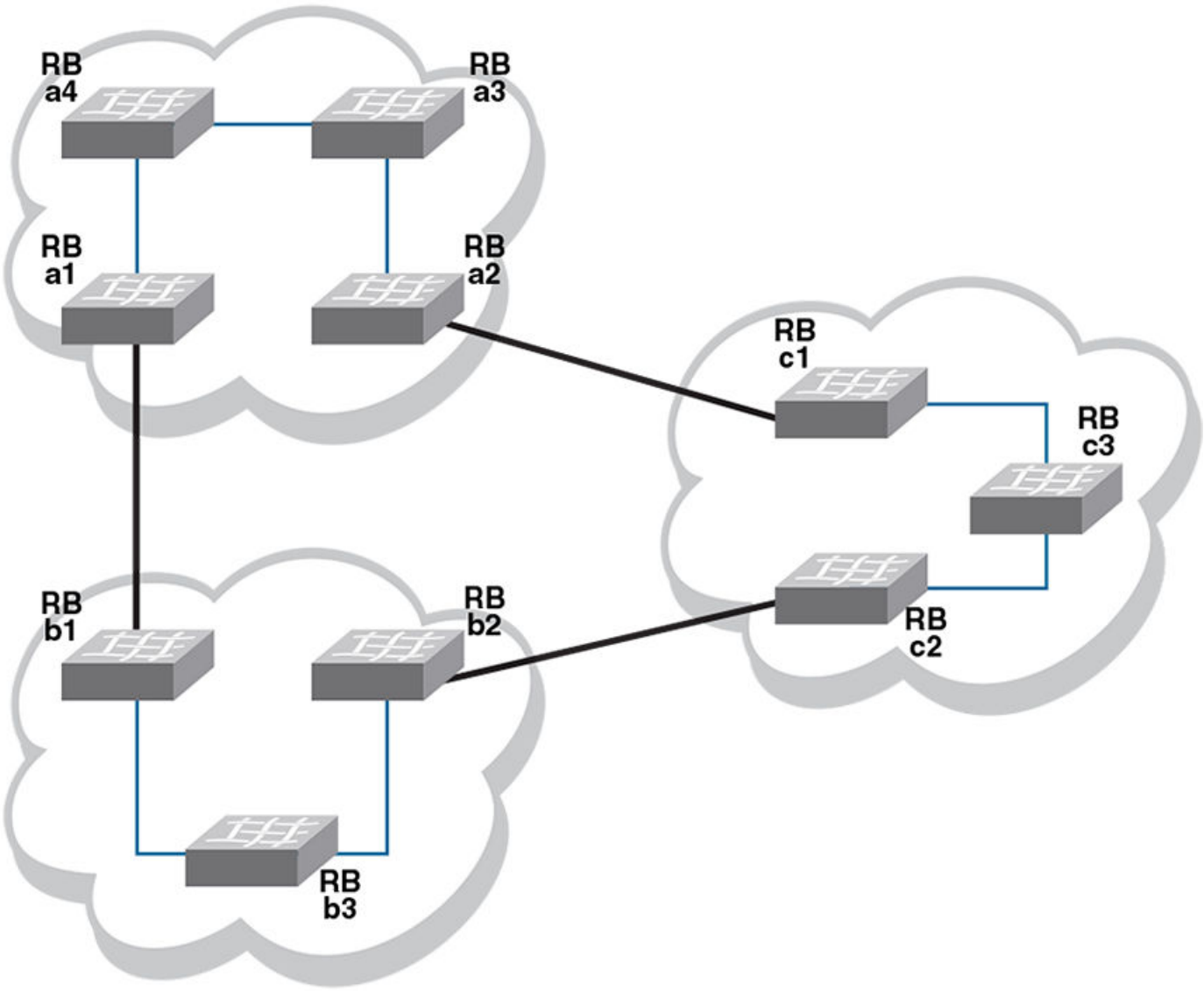
The following figure shows an example of a misconfiguration between a Brocade VCS Fabric cluster and a standalone switch that could cause a Layer 2 loop. In this case, a VLAG is configured on the edge devices of the Brocade VCS Fabric cluster for the two ISLs that connect the Brocade VCS Fabric cluster to the standalone switch. In this case, a LAG has not been created on the standalone switch at the other end of the ISLs. ELD detects and breaks this potential Layer 2 loop.

FIGURE 1 Missing LAG causes loop



The following figure shows another example for which ELD could be used to detect and break a Layer 2 loop. In this case, multiple Brocade VCS Fabric clusters are interconnected in a manner that creates a Layer 2 loop.

FIGURE 2 Interconnected Brocade VCS Fabric clusters cause loop



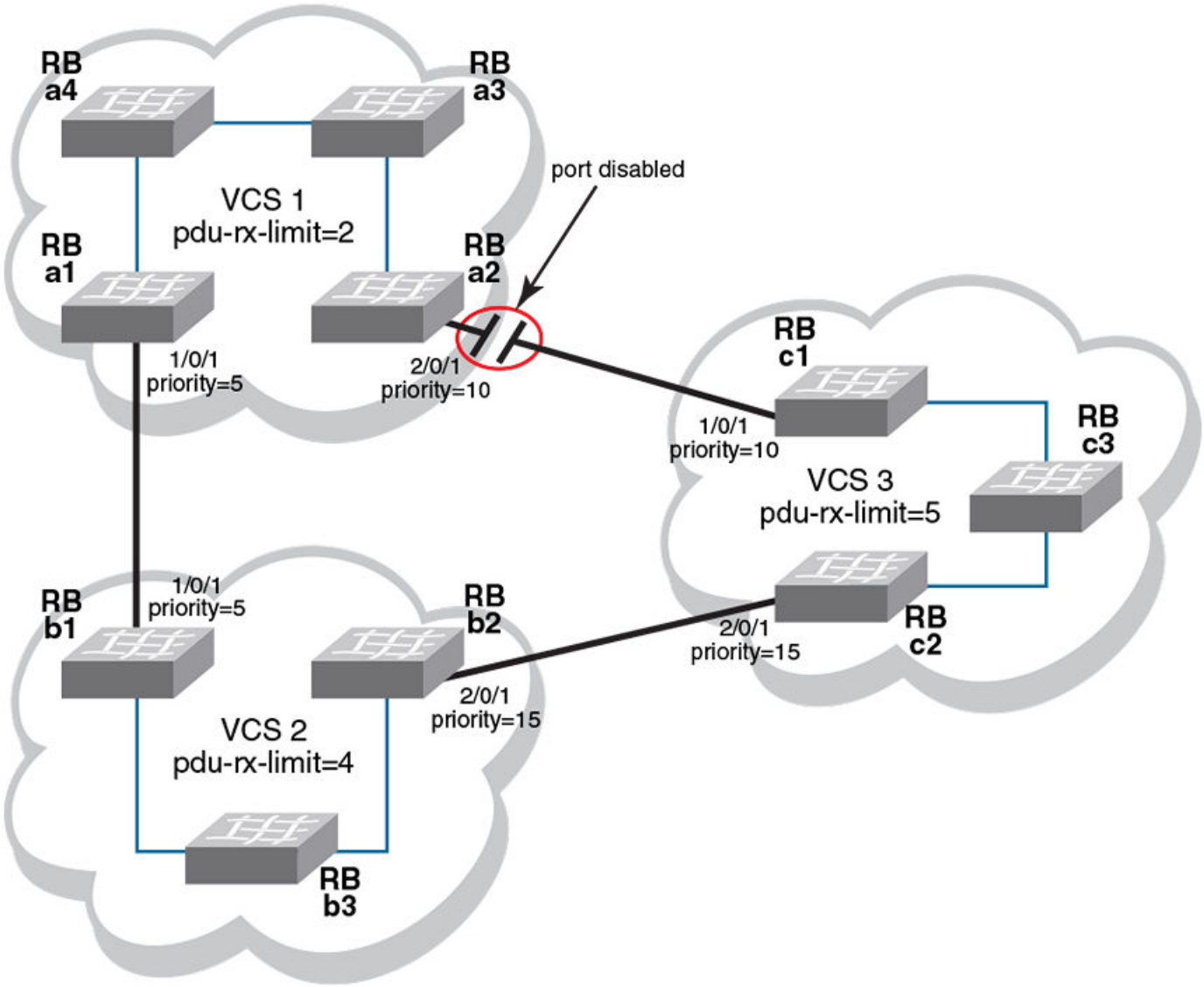
How ELD detects loops

ELD works by multicasting Protocol Data Unit (PDU) packets on edge ports. A device recognizes a loop when it receives a PDU that it initiated. Once the device recognizes that a Layer 2 loop exists, it can take action to disable a port and break the Layer 2 loop.

To minimize the number of disabled ports, ELD assigns a priority to each port and a unique receive limit (pdu-rx-limit) to each Brocade VCS Fabric cluster. The port priority determines whether the sending or receiving edge port of the cluster is disabled. The pdu-rx-limit determines on which Brocade VCS Fabric the action takes place. Without these configured values, it is possible that a Layer 2 loop could be detected in multiple clusters at the same time. As a result, multiple ports would be disabled, stopping traffic among the Brocade VCS Fabric clusters.

The following figure shows the same interconnections as the previous figure under Edge-loop detection overview on page 17, but with ELD enabled on each edge port, and with port priorities and receive limits assigned.

FIGURE 3 Interconnected Brocade VCS Fabric clusters with ELD enabled



With all ELD enabled edge ports sending PDUs at the same rate, VCS1 reaches its pdu-rx-limit first. Port 2/0/1 has a lower priority (higher priority number) than port 1/0/1, and is therefore selected to be disabled. If both ports have the same priority, the port with the higher port-ID is disabled.

If the port being shut down by ELD is part of a LAG, all member ports of the LAG are also shut down. If the port being shut down is part of a vLAG, all member ports of the vLAG on that RBridge are also shut down.

Once ELD disables a port, normal operation is for the port to remain disabled until any misconfiguration is repaired. Once the repair is finished, the port can be re-enabled manually.

NOTE
When ELD disables a port, the port is operationally down, but administratively still up. If a port is disabled by STP or some other Layer 2 protocol, ELD does not process PDUs for that port.

Configuring edge-loop detection

Edge-loop detection requires configuration at the global level and at the interface level. For global level configuration, you need to set the number of PDUs that the Brocade VCS Fabric cluster receives on any port before determining that a loop exists. This value is the **pdu-rx-limit** . You must also set the interval between sending PDUs by using the **hello-interval** command The combination of **pdu-rx-limit** and **hello-interval** determines the time it takes for ELD to detect and break a Layer 2 loop.

At the interface level, you must enable ELD on each port you want it to run on and set the port priority. You should also specify a VLAN on which ELD is enabled

Enter the **pdu-rx-limit** command to set the limit to a different number on each Brocade VCS Fabric cluster so that only one Brocade VCS Fabric cluster disables a port. Set this value in the increment of two to prevent race conditions which might disable ports on two Brocade VCS Fabric clusters that are incrementally only one apart.

Enter the **hello-interval** command to set the interval between PDUs. This interval must be set to the same value on all Brocade VCS Fabric clusters for which ELD is configured, otherwise the results of edge-loop detection become unpredictable.

Optionally, enter the **mac refresh** command to flush MAC addresses at a specified interval (in seconds) on either the entire cluster or on the partner port to remove any MAC inconsistencies in your system. If two interfaces are present in a Layer 2 loop, each interface learns the same set of MAC addresses. When ELD detects the Layer 2 loop, it puts the participating interface into an operationally down state. Consequently, MAC addresses learned on that interface get flushed. However, the same MAC addresses are present at the interface at the other end of the already detected loop, thereby creating this MAC inconsistency.

Optionally, enter the **shutdown-time** command to configure ports to be re-enabled after a specified period of time (range 10 minutes to 24 hours). A typical use for this feature is in environments in which reconfiguration is common, such as in a typical lab environment. Typical use is to allow the default value of zero, which does not allow ports to be re-enabled automatically.

NOTE
Any change to **shutdown-time** only takes effect for the ports that are disabled by ELD after the configuration change. Any ports that were already disabled by ELD before the **shutdown-time** change continues to follow the old **shutdown-time** value. These ports start to follow the new shutdown time after the currently running timer expires and ELD still detects the loop and shuts down the port again.

For each interface on which ELD runs, enter the **edge-loop detection** command to enable ELD. You must also enter the **edge-loop-detection port-priority** command to specify the ELD port priority.

Setting global ELD parameters for a Brocade VCS Fabric cluster

Perform this procedure on every Brocade VCS Fabric cluster where you configure ELD.

The values in this example configure the Brocade VCS Fabric cluster to detect and break loops on receipt of 5 PDUs. Because the PDU interval is set to 2000 ms (2 seconds), any loop breaks after 10 seconds. The selected port will remain disabled for 24 hours, after which it is automatically re-enabled.

1. Log in to any switch in a Brocade VCS Fabric cluster.
2. In global configuration mode, enter the **protocol edge-loop-detection** command to enter edge-loop detection configuration mode.

```
switch(config)# protocol edge-loop-detection
```

3. Enter the **pdu-rx-limit** *number* command to set the number of PDUs that will be received before breaking the Layer 2 loop.

   The *number* variable must be a value in the range1 through 5. The default value is 1.

   ```
   switch(config-eld)# pdu-rx-limit 5
   ```

4. Enter the **hello-interval** *number* command to set the interval between PDUs.

   The *number* variable has a unit of 1 millisecond (ms). It must be in the range from 100 ms to 5000 ms . The default value is 1000 ms.

   ```
   switch(config-eld)# hello-interval 2000
   ```

5. Enter the **mac-refresh** *number* command to flush MAC addresses at a specified interval (in seconds) on either the entire cluster or on the partner port to remove any MAC inconsistencies in your system.

   The *number* variable must be in the range 60 through 300 (seconds). You must also specify either **all** or *port*, depending on whether you want to flush the entire cluster or only the partner port.

   ```
   switch(config-eld)# mac refresh 100 all
   ```

6. Enter the **shutdown-time** *number* command to set the number of minutes after which the shutdown port is re-enabled.

   The *number* variable must be in the range 10 through 1440 (10 minutes through 24 hours). The default value is 0, indicating that the port is not automatically re-enabled.

   ```
   switch(config-eld)# shutdown-time 1440
   ```

Setting interface parameters on a port

Perform this procedure for every port you want to be monitored by ELD.

1. Log in to any switch in a Brocade VCS Fabric cluster.
2. In global configuration mode, enter the **interface** command to select the *rbridge-id/slot/port* for which you want to enable edge-loop detection.
3. In interface configuration mode, enter the **edge-loop-detection vlan** command to specify the VLAN you want ELD to monitor on this port.

   If you do not specify a VLAN, the command fails.
4. Enter the **edge-loop-detection port-priority** command to specify the ELD port priority of the specified port for the selected VLAN. However, enabling switching is not mandatory for assigning a port-priority.

   NOTE
   The priority range of values is from 0 through 255. A port with priority 0 means that shutdown for this port is disabled. The default value port priority is 128

Setting the ELD port priority on two port/VLAN pairs

This example sets the ELD port priority on two port/VLAN pairs: port 1/0/7 VLAN 10 and port 4/0/6 VLAN 10.

If both these ports are detected in the same loop, ELD shuts down port 4/0/6 when the pdu-rx-limit for the Brocade VCS Fabric cluster is reached. Port 4/0/6 is chosen for shut down because it has been assigned the lower priority (higher number) then port 1/0/7.

```
switch(config)# interface TenGigabitEthernet 1/0/7
switch(conf-if-te-1/0/7)# edge-loop-detection vlan 10
switch(conf-if-te-1/0/7)# edge-loop-detection port-priority 5
switch(conf-if-te-1/0/7)# top
```

```
switch(config)# interface TenGigabitEthernet 4/0/6
switch(conf-if-te-1/0/7)# edge-loop-detection vlan 10
switch(conf-if-te-1/0/7)# edge-loop-detection port-priority 7
```

Troubleshooting edge-loop detection

Use the edge-loop detection commands to view and correct misconfigurations

1. Log in to any switch in a Brocade VCS Fabric cluster.
2. In the global configuration mode, enter the **show edge-loop-detection** command to display edge-loop detection statistics for the Brocade VCS Fabric cluster.

   The command output shows ports disabled by ELD.
3. Correct any misconfigurations detected in step 2.
4. Perform one of the following operations in global configuration mode:
   - To re-enable a single port disabled by ELD:
     1. Enter the **shutdown** command on the port.
     2. Enter the **no shutdown** command on the port.

        NOTE
        If an edge-port becomes an ISL port because a remote port's VCS ID was changed, a port that was already shut down by ELD must be cycled with the **shutdown** and **no shutdown** commands to be detected as an ISL port.

   - To re-enable all ports disabled by ELD, enter **clear edge-loop-detection**.

AMPP

AMPP overview

Server virtualization infrastructure associates a server-side Virtual Ethernet Bridge (VEB) port-profile with each Ethernet MAC address used by a virtual machine (VM) to access the network through a VEB port. The Brocade Auto Migrating Port Profile (AMPP) feature provides advanced controls for maintaining and migrating these port-profile associations when a VM migrates across physical servers.

If the VM is migrated from one physical server to another, the VEB's port-profile migrates with it, providing automated port-profile migration of the server's VEB ports that are associated with the VM.

For environments where the server's virtualization infrastructure provides sufficient controls, automated port-profile migration approaches are fine. An example of such an environment is a high performance cluster that uses a Layer 2 network that is isolated from external networks through firewalls and security appliances.

However, there is a gap between the access and Quality of Service (QoS) controls supported in external Layer 2 switches and the server virtualization infrastructure. External Layer 2 switches have more advanced controls compared to server VEB implementations.

Some environments prefer the more advanced controls provided by external network switches. An example of such an environment is a multi-tier data center that has several types of applications, each with differing advanced network controls, running over the same Layer 2 network. In this type of environment, the network administrator often prefers the use of advanced access controls available in external switches.

Layer 2 networks do not provide a mechanism for automatically migrating switch access and traffic controls associated with an end-point device when that device migrates from one switch to another. The migration may be physical, such as an operating system image (such as an application, middleware, operating system, and associated state) that is running BareMetal OS on one system and is migrated to another system. The migration may be also be virtual, such as an operating system image that is running over Hypervisor VMware on one system and is migrated to run over Hypervisor VMware on another system.

AMPP over vLAG

Virtual Link Aggregation Group (vLAG) is the name for Brocade proprietary LAG in which the links to the Brocade VCS Fabric can be connected to one or more physical switches or servers. For redundancy and greater bandwidth, vLAG is a vital component of Brocade VCS Fabric technology. AMPP is supported on vLAG and standard LAG in a manner similar to physical port.

FCoE capability on all port-profiled interfaces can be activated using the fcoe-port configuration in the default port-profile (refer to Configuring FCoE profiles on page 31). This configuration enforces FCoE capability only on physical interfaces, not on the port-channel LAG. Member links of the LAG must be explicitly configured for FCoE capability.

For complete information on vLAG, refer to Link Aggregation Control Protocol on page 177.

The *italic* text in the following example highlights the vLAG information in the port profile:

```
switch# show port-profile status

Port-Profile             PPID     Activated   Associated MAC    Interface
auto-dvPortGroup         1        Yes         None              None
auto-dvPortGroup2        2        Yes         None              None
auto-dvPortGroup3        3        Yes         None              None
auto-dvPortGroup_4_0     4        Yes         0050.567e.98b0    None
auto-dvPortGroup_vlag    5        Yes         0050.5678.eaed    None
auto-for_iscsi           6        Yes         0050.5673.85f9    None
```

```
                                                0050.5673.fc6d    None
                                                0050.5674.f772    None
                                                0050.5675.d6e0    Te 234/0/54
                                                0050.567a.4288    None
auto-VM_Network          9        Yes           000c.2915.4bdc    None
                                                0050.56a0.000d    None
                                                0050.56a0.000e    None
                                                0050.56a0.000f    None
                                                0050.56a0.0010    Po 53
                                                0050.56a0.0011    Po 53
                                                0050.56a0.0012    Po 53
                                                0050.56a0.0013    None
                                                0050.56a0.0025    None
                                                0050.56a0.0026    None
                                                0050.56a0.0027    None
                                                0050.56a0.0028    None
                                                0050.56a0.0029    Po 53
                                                0050.56a0.002a    Po 53
                                                0050.56a0.002b    Po 53
                                                0050.56a0.002c    None
                                                0050.56a0.002d    None
                                                0050.56a0.002e    None
                                                0050.56a0.002f    None
                                                0050.56b3.0001    Po 53
                                                0050.56b3.0002    Po 53
                                                0050.56b3.0004    Po 53
                                                0050.56b3.0005    None
auto-VM_kernel           10       Yes           0050.5671.4d06    None
                                                0050.5672.862f    Po 53
                                                0050.5678.37ea    None
                                                0050.567a.ddc3    None
auto-VM_NW_1G            11       Yes           0050.56b3.0000    None
                                                0050.56b3.0003    Po 82
                                                0050.56b3.0007    None
                                                0050.56b3.0008    Po 82
                                                0050.56b3.0009    Po 82
auto-VMkernel            12       Yes           0050.567a.fdcf    Po 82
                                                0050.567c.c2e3    None
auto-VMkernel_VS         13       Yes           0050.567d.16b9    None
                                                0050.567e.e25b    None
auto-Management+Network  14       Yes           5cf3.fc4d.ca88    None
auto-Virtual+Machine+Network  15  Yes           000c.2941.27e2    None
                                                000c.2980.335d    None

switch# show port-profile int all
Interface        Port-Profile
Gi 234/0/1       None
Gi 234/0/13      None
Gi 234/0/25      None
Gi 234/0/26      None
Te 234/0/54      auto-for_iscsi
Po 82            auto-VM_NW_1G
                 auto-VMkernel
Po 53            auto-VM_Network
                 auto-VM_kernel
```

AMPP and Switched Port Analyzer

Switched Port Analyzer (SPAN), or Port Mirroring, selects network traffic for analysis by a network analyzer. If you are interested in listening or snooping on traffic that passes through a particular port, Port Mirroring is necessary to artificially copy the packets to a port connected to the analyzer. However, SPAN and the Auto Migrating Port Profile (AMPP) port-profile-port configuration are mutually exclusive. All of the Layer 2 configuration for SPAN is mutually exclusive with regard to the **port-profile-port** command for AMPP.

AMPP scalability

The following table describes the Auto Migrating Port Profile (AMPP) scalability values supported by Network OS 5.0.0 and later.

TABLE 1 AMPP scalability values

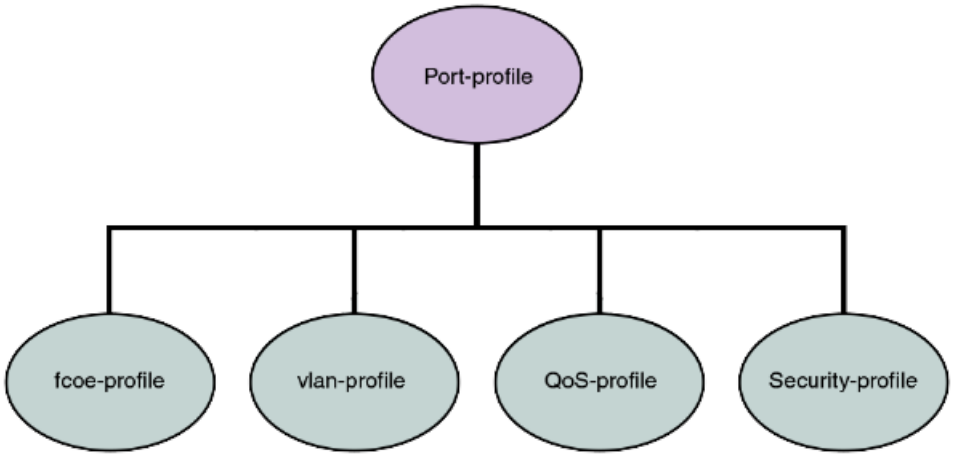
Metric	Logical chassis mode	Fabric cluster mode
Number of profiles	1000	1000
Number of VLANs in port-profiles	2000	2000
QoS profile	1 cee-map 1 mutation-map	1 cee-map 1 cos-mutation-map
Number of ACLs in security profiles	1 ingress MAC ACL 1 egress MAC ACL 1 ingress IPv4 ACL 1 egress IPv4 ACL 1 ingress IPv6 ACL 1 egress IPv6 ACL	1 ingress MAC ACL 1 egress MAC ACL 1 ingress IPv4 ACL 1 egress IPv4 ACL 1 ingress IPv6 ACL 1 egress IPv6 ACL

For the number of MAC associations that are supported, refer to the Release Notes. The practical number of MAC associations and VLANs that are supported will vary depending on the ACL configuration. In addition, AMPP is subject to the maximum number of vLAGs and LAGs supported on the switch, which is 1000 in this case.

AMPP port-profiles

As shown in the following figure, the default port-profile contains the entire configuration needed for a VM to get access to the LAN and SAN.

FIGURE 4 Port-profile contents



In addition, all the combinations can be mixed up with some security rules grouped under a security-profile.

NOTE

A port-profile does not contain some of the interface level configurations, such as LLDP, SPAN, LAG, and so on.

A port-profile operates as a self-contained configuration container. In other words, if a port-profile is applied on a completely new switch without any configuration, it is capable of configuring the interface's local configuration and starting to carry traffic. Any changes to the policies are immediately applied to the data plane.

Security profiles are applied to the ACLs based on the profile or PolicyID. Therefore, multiple security profiles can be applied to the same profiled port.

NOTE

The fcoe-profile is available for both default and nondefault profiles. User-defined port-profiles do not have access to the fcoe-profile. However, editing of the port-profile is not allowed once the port-profile is activated. Activation of the port-profile is mandatory when it is applied to a port. Refer to Configuring FCoE profiles on page 31 for additional details.

Life of a port-profile

A port-profile during creation will go through multiple states. The states of a port-profile are as follows:

- Created —This state specifies that a port-profile is created or modified, but may not be complete.
- Activated —This state specifies that a port-profile is activated and is available for MAC->port-profile association. If the port-profile created is not complete then the activation fails; you must resolve any conflicts or dependencies and reactivate the port-profile.
- Associated —This state specifies that one or more MAC addresses have been associated to this port-profile within the fabric.
- Applied —This state indicates that the port-profile is applied on the profiled port where the associated MAC address appeared. In the absence of any signaling protocol, the system snoops the packet to detect if the associated MAC address has appeared on the profiled port. Configuration of two different port-profiles can co-exist on a profiled port, but if there is a conflict then the application of the later port-profile fails.

The following table describes the AMPP events and the applicable failure behaviors.

TABLE 2 AMPP behavior and failure descriptions

AMPP event	Applicable behavior and failures
Create port-profile	• If the port-profile does not exist, then it is created. If it exists, then it is available for modification (if it is not yet activated).
Activate port-profile	• If the port-profile configuration is not complete, activation fails. Unless the port-profile is activated, it is not applied on any port-profile-port. • If all the dependency validations succeed, the port-profile is in the ACTIVE state and is ready for association. • A vlan-profile is mandatory for all port-profiles.
De-activate port-profile	• This event removes the applied port-profile configuration from all the profiled-ports. • De-activation is allowed even if there are MAC addresses associated with the port-profile.
Modify port-profile	• Port-profile can be edited only in the pre-activation stage. • The port-profile is set to the INACTIVE state if any conflicting attributes are configured, or some dependent configuration is not completed. • Port-profile state is set as INACTIVE and any attempt to associate the port-profile to a MAC address may not be allowed.
Associate MAC addresses to a port-profile	• If the MAC is already associated with a port-profile, the port-profile to MAC association fails. • Otherwise, if the port-profile to MAC association succeeds, when the same MAC is learned on any of the ports, the port-profile which has the MAC association is applied to the port.
De-associate MAC addresses from a port-profile	• If mapping exists, all the policies configured for a specific MAC address are removed from that port or switch.
Deleting a port-profile	• An IN USE error is generated if the port-profile is in an activated state. AMPP forces you to de-activate the profile before deleting.

TABLE 2 AMPP behavior and failure descriptions (continued)

AMPP event	Applicable behavior and failures
	• If the port-profile is in an inactive state, then deletion of profile removes all the MAC associations as well.
Modifying port-profile content when in an associated state	• An IN USE error is generated if the port-profile is already activated.
Moving the VM MAC and notifying the fabric	• All policies associated to the port-profile ID are mapped on the MAC address and applied to the new port in the fabric.
Unused port-profile	• You must manually remove the port-profile to MAC associations.

Configuring AMPP profiles

The following sections cover configuring, deleting, and monitoring various AMPP-related profiles.

Configuring a new port-profile

To support VM MAC address learning, the default port-profile is employed. The default profile is different from the other user-defined AMPP profiles:

- The **allow non-profiled-macs** command only functions with the default port-profile. Refer to the *Network OS Command Reference*.
- The **restrict-flooding** command only functions with the default port-profile. Refer to the *Network OS Command Reference*.
- The port-profile ID (ppid) of the profile cannot be changed.
- The VLAN sub-profile cannot be modified.
- The QoS sub-profile and security-profile cannot be added.
- The default port-profile cannot be activated.

Brocade recommends that you create a new port-profile to accommodate your requirements. To configure a new port-profile, perform the following steps in privileged EXEC mode.

1. Configure the physical interface, LAG, or vLAG as a port-profile port.

```
switch(if-te-2/0/1)# port-profile-port
```

2. Create and configure a new port-profile name.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# port-profile vm1-port-profile
switch(config-port-profile-vm1-port-profile)# vlan-profile
switch(config-pp-vlan)# switchport
switch(config-pp-vlan)# switchport mode trunk
switch(config-pp-vlan)# switchport trunk native-vlan 300
switch(config-pp-vlan)# switchport trunk allowed vlan add 300
```

3. Exit VLAN profile configuration mode.

```
switch(config-pp-vlan)# exit
```

4. Activate the profile.

```
switch(config)# port-profile vm1-port-profile activate
```

5. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vm1-port-profile static 0050.56bf.0001
switch(config)# port-profile vm1-port-profile static 0050.56bf.0002
switch(config)# port-profile vm1-port-profile static 0050.56bf.0003
switch(config)# port-profile vm1-port-profile static 0050.56bf.0004
switch(config)# port-profile vm1-port-profile static 0050.56bf.0005
```

Configuring VLAN profiles

The VLAN profile defines the VLAN membership of the overall port-profile, which includes both the tagged and untagged VLANs.

NOTE
Private VLAN port mode commands are not available for AMPP VLAN profiles.

To configure the VLAN profile, perform the following steps in global configuration mode.

1. AMPP profiles cannot be modified while active. De-activate the port-profile before modifying the VLAN profile.

```
switch(config)# no port-profile vm1-port-profile activate
```

2. Enter VLAN profile configuration mode.

```
switch(config)# port-profile vm1-port-profile
switch(config-port-profile-vm1-port-profile)# vlan-profile
```

3. Use the **switchport** command to change the mode to Layer 2 and set the switching characteristics to the defaults.

```
switch(config-pp-vlan)# switchport
```

4. Access the VLAN profile mode for the correct VLAN.

```
switch(config-pp-vlan)# switchport access vlan 200
```

5. Enter trunk configuration mode.

```
switch(config-pp-vlan)# switchport mode trunk
```

6. Configure the trunk mode for the allowed VLAN IDs.

```
switch(config-pp-vlan)# switchport trunk allowed vlan add 10, 20, 30-40
```

7. Configure the trunk mode to be a native VLAN.

```
switch(config-pp-vlan)# switchport trunk native-vlan 300
```

8. Exit VLAN profile configuration mode.

```
switch(config-pp-vlan)# exit
```

9. Activate the profile.

```
switch(config)# port-profile vm1-port-profile activate
```

10. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vm1-port-profile static 0050.56bf.0001
switch(config)# port-profile vm1-port-profile static 0050.56bf.0002
switch(config)# port-profile vm1-port-profile static 0050.56bf.0003
switch(config)# port-profile vm1-port-profile static 0050.56bf.0004
switch(config)# port-profile vm1-port-profile static 0050.56bf.0005
```

Configuring FCoE profiles

Both default and nondefault port profies are supported. Refer to FCoE on page 37 for details.

> **NOTE**
> Before a port profile can be modified, no interfaces can have a port-profile-port configuration.

In the absence of the FCoE profile in the default AMPP profile, you can configure FCoE on a per-interface basis, based on the profiled ports.

To configure a default FCoE profile globally, perform the following steps in global configuration mode.

1. Enter port-profile configuration mode.

```
switch(config)# port-profile default
```

2. Enter FCoE-profile configuration mode.

```
switch(config-port-profile-default)# fcoe-profile
```

3. Activate the FCoE port profile.

   An FCoE map cannot be applied on interfaces that already have a CEE map applied to it.

```
switch(config-fcoe-profile)# fcoeport default
```

Configuring QoS profiles

QoS profiles define the following values:

- Incoming 802.1p priority is set to internal queue priority. If the port is in QoS untrusted mode, all incoming priorities will be mapped to default best effort priority.
- Incoming priority is set to outgoing priority.
- Mapping of incoming priorities is set to strict or WRR traffic classes.
- Enabling of flow control on a strict or a WRR traffic class.

The QoS profile has two flavors: CEE QoS and Ethernet QoS. The QoS profile may contain either CEE QoS or Ethernet QoS. Server side ports typically are carrying converged traffic.

To configure the QoS profile, perform the following steps in global configuration mode.

1. AMPP profiles cannot be modified while active. Deactivate the port-profile before modifying the VLAN profile.

```
switch(config)# no port-profile vm1-port-profile activate
```

2. Enter QoS profile mode.

```
switch(config)# port-profile vm1-port-profile
switch(config-port-profile-vm1-port-profile)# qos-profile
switch(config-qos-profile)#
```

3. Apply the CEE map.

```
switch(config-qos-profile)# cee default
```

4. Apply a map to the profile. You can do either of the following:

- Apply the existing CoS-to-CoS mutation map.

```
switch(config-qos-profile)# qos cos-mutation vm1-cos2cos-map
```

- Apply the existing CoS-to-Traffic-Class map.

```
switch(config-qos-profile)# qos cos-traffic-class vm1-cos2traffic-map
```

5. Enable pause generation for each CoS.

```
switch(config-qos-profile)# qos flowcontrol tx on rx on
```

6. Exit QoS profile mode.

```
switch(config-qos-profile)# exit
```

7. Activate the profile.

```
switch(config)# port-profile vm1-port-profile activate
```

8. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vm1-port-profile static 0050.56bf.0001
switch(config)# port-profile vm1-port-profile static 0050.56bf.0002
switch(config)# port-profile vm1-port-profile static 0050.56bf.0003
switch(config)# port-profile vm1-port-profile static 0050.56bf.0004
switch(config)# port-profile vm1-port-profile static 0050.56bf.0005
```

Configuring security profiles

A security profile defines all the security rules needed for the server port. A typical security profile contains attributes for MAC-based and IP-based standard and extended ACLs. Security profiles are applied to the ACLs based on the profile or PolicyID. Therefore, multiple security profiles can be applied to the same profiled port.

To configure the security profile, perform the following steps in global configuration mode.

1. AMPP profiles cannot be modified while active. Deactivate the port-profile before modifying the security profile.

```
switch(config)# no port-profile vm1-port-profile activate
```

2. Enter security profile configuration mode.

```
switch(config)# port-profile vm1-port-profile
switch(config-pp)# security-profile
switch(config-pp-security)#
```

3. Modify the ACL security attributes. Refer to the *Network OS Security Guide* for details on modifying ACLs.
4. Apply the ACL to the security profile.

```
switch(config-pp-security)# mac access-group vm1-acl in
```

5. Exit security profile configuration mode.

```
switch(config-pp-security)# exit
```

6. Activate the profile.

```
switch(config)# port-profile vm1-port-profile activate
```

7. Associate the profile to the MAC address for each host.

```
switch(config)# port-profile vm1-port-profile static 0050.56bf.0001
switch(config)# port-profile vm1-port-profile static 0050.56bf.0002
switch(config)# port-profile vm1-port-profile static 0050.56bf.0003
switch(config)# port-profile vm1-port-profile static 0050.56bf.0004
switch(config)# port-profile vm1-port-profile static 0050.56bf.0005
```

Creating a port-profile-port

To create a port-profile-port, perform the following steps in global configuration mode.

1. Activate the interface configuration mode for the interface you wish to modify.

   The following example activates the mode for the 10-gigabit Ethernet interface in slot 0/port 0.

```
switch(config)# interface tengigabitethernet 1/0/1
```

2. Configure port-profile-port on the physical interface.

```
switch(conf-int-te-1/0/1)# port-profile-port
```

Deleting a port-profile-port

To delete a port-profile-port, perform the following steps in global configuration mode.

1. Activate the interface configuration mode for the interface you wish to modify.

   The following example activates the mode for the 10-gigabit Ethernet interface in slot 0/port 0.

```
switch(config)# interface tengigabitethernet 1/0/1
```

2. Unconfigure port-profile-port on the physical interface.

```
switch(conf-int-te-1/0/1)# no port-profile-port
switch(conf-int-te-1/0/1)# no shutdown
```

Deleting a port-profile

To delete a port-profile, perform the following steps in privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

2. Deactivate the port-profile.

```
switch(config)# no port-profile vm1-port-profile activate
```

3. Use the **no** form of the **port-profile** command to delete the custom profile.

   You cannot delete the default port-profile.

```
switch(config)# no port-profile vm1-port-profile
```

Deleting a sub-profile

To delete a sub-profile, perform the following steps in privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

2. Deactivate the port-profile.

```
switch(config)# no port-profile vm1-port-profile activate
```

3. Enter port-profile mode.

```
switch(conf-vm1-port-profile)# port-profile vm1-port-profile
```

4. Use the following to delete sub-profiles.
   - To delete a VLAN sub-profile:

```
switch(conf-vm1-port-profile)# no vlan-profile
```

   - To delete a security sub-profile:

```
switch(conf-vm1-port-profile)# no security-profile
```

   - To delete a FCoE sub-profile under default profile:

```
switch(conf-pp-default)# no fcoe-profile
```

   - To delete a QoS sub-profile:

```
switch(conf-vm1-port-profile)# no qos-profile
```

Creating a new port-profile domain and adding port profiles

Use the **port-profile-domain** command to create an AMPP port-profile domain that contains all of the port profiles that can be applied to a profiled port in a Virtual Fabrics context.

1. In global configuration mode, create a port-profile domain.

```
device(config)# port-profile-domain my_PP_domain
```

2. In port-profile-domain configuration mode, use the **port-profile** command to add profiles to that domain.

```
device(config-port-profile-domain-my_PP_domain)# port-profile my_PP_1
device(config-port-profile-domain-my_PP_domain)# port-profile my_PP_2
```

Monitoring AMPP profiles

To monitor the AMPP profiles, perform the following steps in privileged EXEC mode.

1. Use the **show** command to display the current MAC details.

```
switch# show mac-address-table port-profile
Legend: Untagged(U), Tagged (T), Not Forwardable(NF) and Conflict(C)
VlanId  Mac-address     Type     State    Port-Profile   Ports
1       0050.5679.5351  Dynamic  Active   Profiled(U)    Te 111/0/10
1       0050.567b.7030  Dynamic  Active   Profiled(U)    Te 111/0/12
1       005a.8402.0000  Dynamic  Active   Profiled(T)    Te 111/0/24
```

```
1       005a.8402.0001  Dynamic  Active   Profiled(NF)   Te 111/0/24
1       005a.8402.0002  Dynamic  Active   Not Profiled   Te 111/0/24
1       005a.8402.0003  Dynamic  Active   Not Profiled   Te 111/0/24
1       005a.8402.0004  Dynamic  Active   Not Profiled   Te 111/0/24
1       005a.8402.0005  Dynamic  Active   Profiled(NF)   Te 111/0/24
1       005a.8402.0006  Dynamic  Active   Not Profiled   Te 111/0/24
1       005a.8402.0007  Dynamic  Active   Profiled(T)    Te 111/0/24
1       005b.8402.0001  Dynamic  Active   Profiled(T)    Te 111/0/24
1       005c.8402.0001  Dynamic  Active   Profiled(T)    Te 111/0/24
100     005a.8402.0000  Dynamic  Active   Profiled       Te 111/0/24
100     005a.8402.0001  Dynamic  Active   Profiled(NF)   Te 111/0/24
100     005a.8402.0003  Dynamic  Active   Not Profiled   Te 111/0/24
100     005a.8402.0005  Dynamic  Active   Profiled(NF)   Te 111/0/24
100     005a.8402.0007  Dynamic  Active   Profiled       Te 111/0/24
Total MAC addresses  : 17
```

2. Use the **show running-config** command to display all the available port-profile configurations.

```
switch# show running-config port-profile
port-profile default
 vlan-profile
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
  switchport trunk native-vlan 1
 !
 fcoe-profile
  fcoeport default
 !
!
port-profile pp1
 vlan-profile
  switchport
  switchport mode access
  switchport access vlan 1
 !
 qos-profile
 !
!
port-profile pp1 activate
port-profile pp1 static 1000.0000.0001
```

3. Use the **show port-profile** command to display the current port-profile configuration.

```
switch# show port-profile
port-profile default
ppid 0
vlan-profile
  switchport
  switchport mode trunk
  switchport trunk native-vlan 1
port-profile UpgradedVlanProfile
ppid 1
vlan-profile
  switchport
  switchport mode trunk
  switchport trunk allowed vlan all
```

4. Use the **show port-profile status** command to display the current status of all AMPP profiles.

```
switch# show port-profile status applied
Port-Profile      PPID   Activated   Associated MAC  Interface
auto-for_iscsi    6      Yes         0050.5675.d6e0  Te 9/0/54
auto-VM_Network   9      Yes         0050.56b3.0001  Te 9/0/53
                                     0050.56b3.0002  Te 9/0/53
                                     0050.56b3.0004  Te 9/0/53
                                     0050.56b3.0014  Te 9/0/53
switch# show port-profile status activated
Port-Profile           PPID   Activated   Associated MAC  Interface
auto-dvPortGroup       1      Yes         None            None
```

```
auto-dvPortGroup2      2      Yes         None            None
auto-dvPortGroup3      3      Yes         None            None
auto-dvPortGroup_4_0   4      Yes         0050.567e.98b0  None
auto-dvPortGroup_vlag  5      Yes         0050.5678.eaed  None
auto-for_iscsi         6      Yes         0050.5673.85f9  None
switch# show port-profile status associated
Port-Profile           PPID   Activated   Associated MAC  Interface
auto-dvPortGroup_4_0   4      Yes         0050.567e.98b0  None
auto-dvPortGroup_vlag  5      Yes         0050.5678.eaed  None
auto-for_iscsi         6      Yes         0050.5673.85f9  None
```

5. Use show port-profile interface all to display profile and applied interface information.

```
switch# show port-profile interface all
Port-profile      Interface
auto-VM_Network   Te 9/0/53
auto-for_iscsi    Te 9/0/54
```

FCoE

FCoE overview

Fibre Channel over Ethernet (FCoE) enables you to transport FC protocols and frames over Data Center Bridging (DCB) networks. DCB is an enhanced Ethernet network that enables the convergence of various applications in data centers (LAN, SAN, and HPC) onto a single interconnect technology.

FCoE provides a method of encapsulating the Fibre Channel (FC) traffic over a physical Ethernet link. FCoE frames use a unique EtherType [FCoE uses 0x8906 and FCoE Initialization Protocol (FIP) uses 0x8914] that enables FCoE SAN traffic and legacy LAN Ethernet traffic to be carried on the same link. FC frames are encapsulated in an Ethernet frame and sent from one FCoE-aware device across an Ethernet network to a second FCoE-aware device. The FCoE-aware devices may be FCoE end nodes (ENodes) such as servers, storage arrays, or tape drives on one end and FCoE Forwarders on the other end. FCoE Forwarders (FCFs) are switches providing SAN fabric services and may also provide FCoE-to-FC bridging.

The motivation behind using DCB networks as a transport mechanism for FC arises from the desire to simplify host protocol stacks and consolidate network interfaces in data center environments. FC standards allow for building highly reliable, high-performance fabrics for shared storage, and these characteristics are what DCB brings to data centers. Therefore, it is logical to consider transporting FC protocols over a reliable DCB network in such a way that it is completely transparent to the applications. The underlying DCB fabric is highly reliable and high performing, the same as the FC SAN.

In FCoE, ENodes discover FCFs and initialize the FCoE connection through the FCoE Initialization Protocol (FIP). FIP has a separate EtherType from FCoE. FIP includes a discovery phase in which ENodes discover VLANs supporting FCoE, solicit FCFs on those VLANs, and FCFs respond to the solicitations with advertisements of their own. At this point, the ENodes know enough about the FCFs to log in to them. The virtual link establishment and fabric login (FLOGI/FDISC) for VN-to-VF port connections is also part of FIP.

Network OS supports the following:

- 100-Gbps blades
- 40-Gbps breakout Inter-Switch Links (ISLs)
- Changes to the way in which the number of FCoE interfaces are created, through the **fcoe-enodes** command
- FCoE logical SANs
- FCoE troubleshooting commands

FCoE terminology

The following table lists and describes the FCoE terminology used in this document.

TABLE 3 FCoE terminology

Term	Description
FCoE	Fibre Channel over Ethernet
DCB	Data Center Bridging
VN_Port	FCoE equivalent of an FC N_Port
VF_Port	FCoE equivalent of an FC F_Port

TABLE 3 FCoE terminology (continued)

Term	Description
ENode	An FCoE device that supports FCoE VN_Ports (servers and target devices)

End-to-end FCoE

The Brocade VCS Fabric is a convergence-ready fabric. This means it is capable of providing lossless service and other features expected of a CEE-capable network. This includes support for multi-hop FCoE, where an FCoE initiator can communicate with an FCoE target that is a number of hops away.

FCoE operations

Each switch in the Brocade VCS Fabric cluster acts as a fully functional FCoE Forwarder (FCF). All Fibre Channel (FC) services required to support a Virtual Network (VN) must run on every Brocade VCS Fabric cluster switch, and each switch in the fabric acts as if it were a separate domain in an FC SAN.

For all practical purposes, a Brocade VCS Fabric operates similarly to an FC fabric because all the FCoE initiators and targets are connected to the Brocade VCS Fabric. Each switch in the cluster gets a domain ID, and once the fabric forms, all the FC services (such as Name Server, Login Controller, Domain Controller) are available on each individual cluster switch.

Network OS 4.0.0 and later supports FCR/LSAN zoning. A combination of 2000 FCoE devices and 1000 FC routed devices (for a total maximum of 3000 devices) is the fabric limit. Because open zoning floods all the State Change Notifications (SCNs) to every device, it should be used only when the fabric has 300 total devices or fewer. Fabrics with higher device counts should have user-defined zoning configurations, with a maximum of 255 devices per zone.

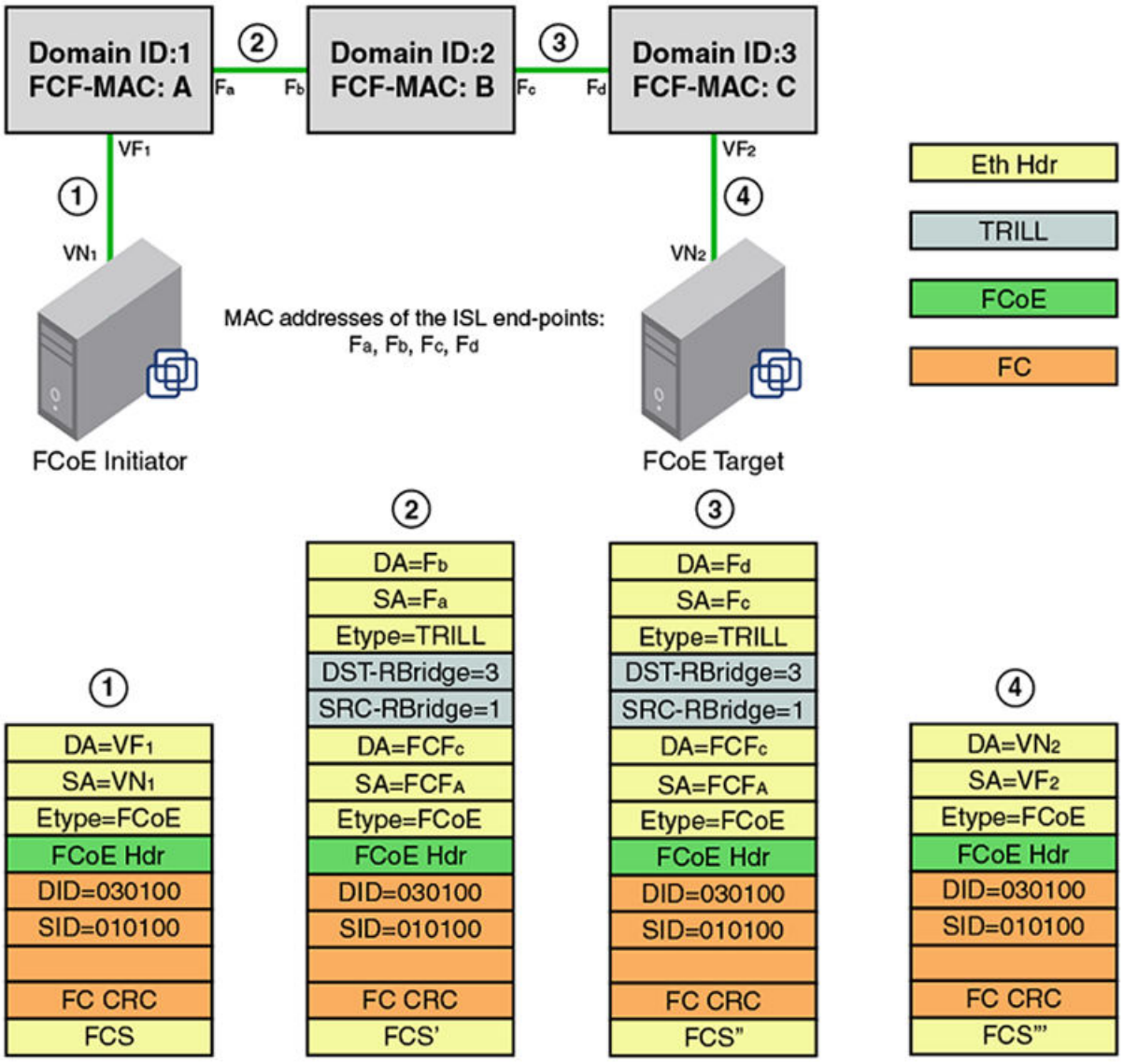
FCoE traffic forwarding across the fabric follows the same equal-cost multi-path (ECMP) routing rules as does LAN traffic forwarding.

FCoE end-to-end forwarding

FCoE frame forwarding between two FCoE devices attached to the Brocade VCS Fabric works similarly to Layer 3 IP routing. The end-node talks to the default gateway's MAC address and the Layer 2 headers are modified hop-by-hop until the frame reaches its final destination. Forwarding decisions are based on the contents of the IP header in the case of IP routing, and the IP header is untouched along the path. FCoE forwarding works the same way.

The following figure illustrates this process. Assume that VN1 (an FCoE initiator) is trying to access VN2 (an FCoE target).

FIGURE 5 FCoE end-to-end header process



1. VN1 and VN2 discover VF1 and VF2 through FIP Discovery Protocol and perform a Fabric Login (FLOGI) to their respective VF ports. That is, VN1 performs an FIP FLOGI to VF1 and VN2 performs a FIP FLOGI to VF2. This works like IP in that all communication between the end-station and the network happens to the router's MAC address at Layer 2. This means VN1 is always communicating with VF1 at Layer 2.
2. In a Brocade VCS Fabric implementation, all FC services are available on every cluster unit. This means there is Fibre Channel Network Switch (FCNS) available on both FCF1 and FCF2. The FCNS service functions identically as it does in an FC SAN. As a result, VN1 discovers VN2.
3. VN1 attempts an N_Port Login (PLOGI) to VN2, with the frame information shown at point 1 in the following figure. The Layer 2 header contains VF1 as the destination MAC address. The Layer 3 header (in this case, the FC header) contains the actual DID and SID of the initiator and the target respectively.

In this example, because VN1 is connected to the FCF with a Domain ID of 1, its PID is 010100. Similarly, because VN2 is connected to FCF3, its FC address is 030100.

4. When FCF-A receives the frame on VF1, it performs a Layer 3 lookup. It looks up the DID in the FC header and determines that the frame is destined to a non-local domain. FCF-A decodes the next hop needed to reach the destination domain of 3, based on Fabric Shortest Path First (FSPF). It is at this point that it does something different than a normal IP router.
5. FCF-A now knows that it needs to reach FCF-C. Each FCF in the Brocade VCS Fabric is assigned an FCF MAC address. FCF-A constructs the Layer 2 header based on this information. So, the original MAC header is now transformed as follows: the DA is changed from VF1 to FCF-C and the SA is changed from VN1 to FCF-A. This occurs at point 2 in the above figure.
6. The frame gets a Transparent Interconnection of Lots of Links (TRILL) header and traverses across the fabric to reach FCF-C. The TRILL header indicates that the source is RBridge 1 and the destination is RBridge 3. This occurs at point 2 in the above figure.
7. The outer MAC header is a link level header that gets the frame from FCF-A to FCF-B. FCF-B receives the frame. FCF-B scans the TRILL header, decodes the destination RBridge ID in the frame, and forwards the frame. FCF-B only modifies the Layer 2 header. It neither looks up nor modifies anything in the FC header or the inner MAC header. This occurs at point 3 in the above figure.
8. FCF-C receives the frame. FCF-C scans the TRILL header and decodes the destination RBridge ID. FCF-C promotes the frame to Layer 3 lookup, because the FCF-C is the DA in the inner MAC header. FCF-C then scans the FC header and does something similar to an area route lookup in FC SAN. This lookup yields the MAC address of VN2 and the VF interface (in this case, VF2) information that it needs to use to forward the frame to VN2. This is occurs at point 4 in the above figure.
9. VN2 receives the PLOGI. The PLOGI response from VN2 traverses back to VN1 in similar fashion.

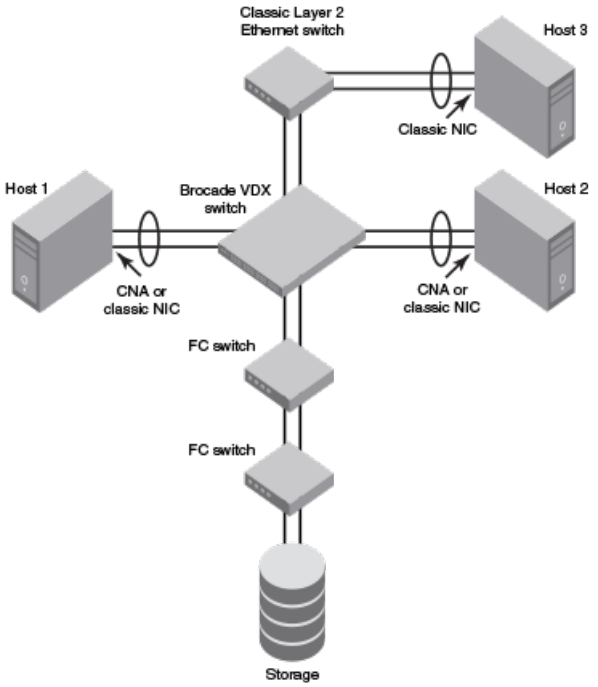
NOTE

It is assumed that both VN1 and VN2 are configured to be in the same FCoE VLAN, and FCoE forwarding is enabled on this VLAN in the Brocade VCS Fabric. Network OS v4.0.0 and later supports only one FCoE VLAN for all FCoE devices connected to the fabric.

FCoE and Layer 2 Ethernet

The Brocade VDX hardware contains DCB ports that support FCoE forwarding. The DCB ports are also backwards-compatible and support classic Layer 2 Ethernet networks (as shown in the following figure). In Layer 2 Ethernet operation, a host with a Converged Network Adapter (CNA) can be directly attached to a DCB port on the Brocade VDX hardware. Another host with a classic 10-gigabit Ethernet network interface card (NIC) can be either directly attached to a DCB port, or attached to a classic Layer 2 Ethernet network that is attached to the Brocade VDX hardware.

FIGURE 6 Multiple switch fabric configuration



Layer 2 forwarding

Layer 2 Ethernet frames are forwarded on the DCB ports. 802.1Q VLAN support is used to tag incoming frames to specific VLANs, and 802.3ac VLAN tagging support is used to accept VLAN tagged frames from external devices.

Network OS uses the following 802.1D bridging protocols between Layer 2 switches and to maintain a loop-free network environment:

- Spanning Tree Protocol (STP)
- Rapid Spanning Tree Protocol (RSTP)
- Multiple Spanning Tree Protocol (MSTP)
- Per-VLAN Spanning Tree (PVST+)
- Rapid Per-VLAN Spanning Tree (RPVST+)

For detailed information on configuring these protocols, refer to STP-Type Protocols on page 147.

The Brocade VDX hardware handles Ethernet frames as follows:

- When the destination MAC address is not in the lookup table, the frame is flooded on all ports in the same VLAN, except the ingress port.
- When the destination MAC address is present in the lookup table, the frame is switched only to the correct egress port.
- When the destination MAC address is present in the lookup table, and the egress port is the same as the ingress port, the frame is dropped.
- If the Ethernet Frame Check Sequence (FCS) is incorrect, because the switch is in cut-through mode, a correctly formatted Ethernet frame is sent out with an incorrect FCS.
- If the Ethernet frame is too short, the frame is discarded and the error counter is incremented.
- If the Ethernet frame is too long, the frame is truncated and the error counter is incremented. The truncated frame is sent out with an incorrect FCS.

- Frames sent to a broadcast destination MAC address are flooded on all ports in the same VLAN, except the ingress port.
- When MAC address entries in the lookup table time out, they are removed. In this event, frame forwarding changes from unicast to flood.
- An existing MAC address entry in the lookup table is discarded when a device is moved to a new location. When a device is moved, the ingress frame from the new port causes the old lookup table entry to be discarded and the new entry to be inserted into the lookup table. Frame forwarding remains unicast to the new port.
- When the lookup table is full, new entries replace the oldest MAC addresses after the oldest MAC addresses reach a certain age and time out. MAC addresses that still have traffic running are not timed out.
- If the port is receiving jumbo frame packets and the port is not configured with the required MTU size to support jumbo frames, then the port discards those frames and increments the over-sized packet error counter.
- If the port is receiving valid multicast frames and the port is **not** part of a VLAN that is enabled for IGMP snooping, then the frames are treated as broadcast frames.
- If the port is receiving multicast frames with a destination MAC address (multicast MAC address) and destination IP address (multicast IP address) belonging to different group addresses or not pointing to the same group, the frames are silently discarded by the port.

NOTE
New entries start replacing older entries when the lookup table reaches 90 percent of its 32Kb capacity.

802.1Q VLAN tagging

The Layer 2 switch always tags an incoming frame with an 802.1Q VLAN ID. If the incoming frame is untagged, then a tag is added according to the port configuration. A port can classify untagged traffic to a single VLAN or to multiple VLANs. If the incoming frame is already tagged, then the port will either forward or discard the frame according to allowed VLAN rules in the port configuration.

These are three examples of 802.1Q VLAN tagging:

- If the DCB port is configured to tag incoming frames with a single VLAN ID, then incoming frames that are untagged are tagged with the VLAN ID.
- If the DCB port is configured to tag incoming frames with multiple VLAN IDs, then incoming frames that are untagged are tagged with the correct VLAN ID based on the port setting.
- If the DCB port is configured to accept externally tagged frames, then incoming frames that are tagged with a VLAN ID are passed through unchanged.

NOTE
Only a single switch-wide VLAN is capable of forwarding FCoE traffic.

For detailed information on configuring VLANs, refer to 802.1Q VLANs on page 71.

Support for Virtual Fabrics

Network OS provides a Virtual Fabrics feature that supports multitenancy by extending the standard (802.1Q) VLAN ID space from 4096 through 8191, enabling the use of classified VLANs. Following an upgrade to Network OS 4.1, the system operates in native VLAN mode until the Virtual Fabrics feature is enabled. In this release, FCoE VLANs are limited to the 802.1Q range of 1 through 4096. FCoE frames are now able to accommodate 802.1AD S-TAGs (service provider tags) and C-TAGs (customer tags) for future support. A C-TAG used to classify an FCoE frame is the same as the VLAN ID and is system wide.

NOTE
Currently, FCoE VLANs can be only 802.1Q VLANs. They cannot be classified or used as C-TAGs for other VLAN classification. All tenant FCoE traffic rides on the same default FCoE VLAN (1002) as in the previous Network OS releases.

For details of the Virtual Fabrics feature, refer to Virtual Fabrics on page 103.

Incoming frame classification

The Brocade VDX hardware is capable of classifying incoming Ethernet frames based on the following criteria:

- Port number
- Protocol
- MAC address

The classified frames can be tagged with a VLAN ID or with 802.1p Ethernet priority. The 802.1p Ethernet priority tagging is done using the Layer 2 Class of Service (CoS). The 802.1p Ethernet priority is used to tag frames in a VLAN with a Layer 2 CoS to prioritize traffic in the VLAN. The Brocade VDX hardware also accepts frames that have been tagged by an external device.

Frame classification options are as follows:

- VLAN ID and Layer 2 CoS by physical port number — With this option, the port is set to classify incoming frames to a preset VLAN ID and the Layer 2 CoS on a physical port on the Brocade VDX hardware.
- VLAN ID and Layer 2 CoS by LAG virtual port number — With this option, the port is set to classify incoming frames to a preset VLAN ID and Layer 2 CoS on a Link Aggregation Group (LAG) virtual port.
- Layer 2 CoS mutation — With this option, the port is set to change the Layer 2 CoS setting by enabling the QoS mutation feature.
- Layer 2 CoS trust — With this option, the port is set to accept the Layer 2 CoS of incoming frames by enabling the QoS trust feature.

For detailed information on configuring QoS, refer to QoS on page 207.

Congestion control and queuing

The Brocade VDX hardware supports several congestion control and queuing strategies. As an output queue approaches congestion, Random Early Detection (RED) is used to selectively and proactively drop frames to maintain maximum link utilization. Incoming frames are classified into priority queues based on the Layer 2 CoS setting of the incoming frame, or the possible rewriting of the Layer 2 CoS field based on the settings of the DCB port or VLAN.

The Brocade VDX hardware supports a combination of two scheduling strategies to queue frames to the egress ports: Priority queuing, which is also referred to as strict priority, and Deficit Weighted Round Robin (DWRR) queuing.

The scheduling algorithms work on the eight traffic classes as specified in 802.1Qaz Enhanced Transmission Selection (ETS).

Queuing features are described as follows:

- RED — RED increases link utilization. When multiple inbound TCP traffic streams are switched to the same outbound port, and some traffic streams send small frames while other traffic streams send large frames, link utilization will not be able to reach 100 percent. When RED is enabled, link utilization approaches 100 percent.
- Classification — Setting user priority.
  - Inbound frames — Inbound frames are tagged with the user priority set for the inbound port. The tag is visible when examining the frames on the outbound port. By default, all frames are tagged to priority zero.
  - Externally tagged Layer 2 frames — When the port is set to accept externally tagged Layer 2 frames, the user priority is set to the Layer 2 CoS of the inbound frames.
- Queuing
  - Input queuing — Input queuing optimizes the traffic flow in the following way. A DCB port has inbound traffic that is tagged with several priority values, and traffic from different priority settings is switched to different outbound ports. Some

outbound ports are already congested with background traffic while others are uncongested. With input queuing, the traffic rate of the traffic streams switched to uncongested ports should remain high.

- Output queuing — Output queuing optimizes the traffic flow in the following way. Several ports carry inbound traffic with different priority settings. Traffic from all ports is switched to the same outbound port. If the inbound ports have different traffic rates, some outbound priority groups will be congested while others can remain uncongested. With output queuing, the traffic rate of the traffic streams that are uncongested should remain high.
- Multicast rate limit — A typical multicast rate limiting example is where several ports carry multicast inbound traffic that is tagged with several priority values. Traffic with different priority settings is switched to different outbound ports. The multicast rate limit is set so that the total multicast traffic rate on output ports is less than the specified set rate limit. Multicast rate-limiting commands are not supported on the Brocade VDX 6740 or VDX 8770. On the latter platforms, use BUM storm control instead.
- Multicast input queuing — A typical multicast input queuing example is where several ports carry multicast inbound traffic that is tagged with several priority values. Traffic with different priority settings is switched to different outbound ports. Some outbound ports are already congested with background traffic while others are uncongested. The traffic rate of the traffic streams switched to the uncongested ports should remain high. All outbound ports should carry some multicast frames from all inbound ports. This enables multicast traffic distribution relative to the set threshold values.
- Multicast output queuing — A typical multicast output queuing example is where several ports carry multicast inbound traffic. Each port has a different priority setting. Traffic from all ports is switched to the same outbound port. If the inbound ports have varying traffic rates, some outbound priority groups will be congested while others remain uncongested. The traffic rate of the traffic streams that are uncongested remains high. The outbound ports should carry some multicast frames from all the inbound ports.

- Scheduling — A typical example of scheduling policy (using Strict Priority 0 and Strict Priority 1 modes) is where ports 0 through 7 carry inbound traffic, each port has a unique priority level, port 0 has priority 0, port 1 has priority 1, and so on. All traffic is switched to the same outbound port. In Strict Priority 0 mode, all ports have DWRR scheduling; therefore, the frames per second (FPS) on all ports should correspond to the DWRR settings. In Strict Priority 1 mode, priority 7 traffic uses Strict Priority; therefore, priority 7 can achieve a higher FPS. Frames from input ports with the same priority level should be scheduled in a round robin manner to the output port.

  When setting the scheduling policy, each priority group that is using DWRR scheduling can be set to use a percentage of the total bandwidth by setting the PG_Percentage parameter.

For detailed information on configuring QoS, refer to QoS on page 207.

Access control

Access Control Lists (ACLs) are used for Layer 2 switching security. Standard ACLs inspect the source address for the inbound ports. Extended ACLs provide filtering by source and destination addresses and protocol. ACLs can be applied to the DCB ports or to VLANs.

ACLs function as follows:

- A standard Ethernet ACL configured on a physical port is used to permit or deny frames based on the source MAC address. The default is to permit all frames.
- An extended Ethernet ACL configured on a physical port is used to permit or deny frames based on the source MAC address, destination MAC address, and EtherType. The default is to permit all frames.
- A standard Ethernet ACL configured on a LAG virtual port is used to permit or deny frames based on the source MAC address. The default is to permit all frames. LAG ACLs apply to all ports in the LAG.
- An extended Ethernet ACL configured on a LAG virtual port is used to permit or deny frames based on the source MAC address, destination MAC address, and EtherType. The default is to permit all frames. LAG ACLs apply to all ports in the LAG.
- A standard Ethernet ACL configured on a VLAN is used to permit or deny frames based on the source MAC address. The default is to permit all frames. VLAN ACLs apply to the Switched Virtual Interface (SVI) for the VLAN.

- An extended Ethernet ACL configured on a VLAN is used to permit or deny frames based on the source MAC address, destination MAC address, and EtherType. The default is to permit all frames. VLAN ACLs apply to the Switched Virtual Interface (SVI) for the VLAN.

For detailed information on configuring ACLs, refer to the "Configuring and Managing ACLs" section of the *Network OS Security Configuration Guide*.

Trunking

NOTE
The term "trunking" in an Ethernet network refers to the use of multiple network links (ports) in parallel to increase the link speed beyond the limits of any one single link or port, and to increase the redundancy for higher availability.

802.1ab Link Layer Discovery Protocol (LLDP) is used to detect links to connected switches or hosts. Trunks can then be configured between an adjacent switch or host and the Brocade VDX hardware.

The Data Center Bridging Capability Exchange Protocol (DCBX) extension is used to identify a DCB-capable port on an adjacent switch or host. For detailed information on configuring LLDP and DCBX, refer to LLDP on page 195.

The 802.3ad Link Aggregation Control Protocol (LACP) is used to combine multiple links to create a trunk with the combined bandwidth of all the individual links. For detailed information on configuring LACP, refer to Link Aggregation on page 177.

Flow control

802.3x Ethernet pause and Ethernet Priority-based Flow Control (PFC) are used to prevent dropped frames by slowing traffic at the source end of a link. When a port on a switch or host is not ready to receive more traffic from the source, perhaps due to congestion, it sends pause frames to the source to pause the traffic flow. When the congestion has been cleared, it stops requesting the source to pause traffic flow, and traffic resumes without any frame drop.

NOTE
Ethernet pause differs from PFC in that the former is applied to all traffic streams irrespective of their COS values, whereas the latter is always applied to a specific COS or priority value.

When Ethernet pause is enabled, pause frames are sent to the traffic source. Similarly, when PFC is enabled, there is no frame drop; pause frames are sent to the source switch.

For detailed information on configuring Ethernet pause and PFC, refer to Configuring QoS on page 220.

Support for 40-Gbps ISLs on breakout ports

40-gigabit-per-second ISLs are supported, including on breakout ports.

FCoE Initialization Protocol

The FCoE Initialization Protocol (FIP) discovers and establishes virtual links between FCoE-capable entities connected to an Ethernet cloud through a dedicated EtherType (0x8914) in the Ethernet frame.

FIP discovery

NOTE
ANSI INCITS 462-2010 Fibre Channel - Backbone - 5 (FC-BB-5) / 13-May-2010 is supported.

The Brocade VDX hardware FIP discovery phase operates as follows:

- The Brocade VDX hardware uses the FCoE Initialization Protocol (FIP). ENodes discover VLANs supporting FCoE, FCFs, and then initialize the FCoE connection through the FIP.
- VF_Port configuration — An FCoE port accepts ENode requests when it is configured as a VF_Port and enabled. An FCoE port does not accept ENode requests when disabled.
- Solicited advertisements — A typical scenario is where a Brocade VDX hardware receives a FIP solicitation from an ENode. Replies to the original FIP solicitation are sent to the MAC address embedded in the original FIP solicitation. After being accepted, the ENode is added to the VN_Port table.
- VLAN 1 — The Brocade VDX hardware should not forward FIP frames on VLAN 1 because it is reserved for management traffic only.
- A fabric-provided MAC address is supported.

NOTE
In the fabric-provided MAC address format, VN_Port MAC addresses are based on a 48-bit fabric-supplied value. The first three bytes of this value are referred to as the FCMAP. The next three bytes are the FC ID, which is assigned by the switch when the ENode logs in to the switch.

FIP login

FIP login operates as follows:

- ENodes can log in to the Brocade VDX hardware using FIP, Fabric login (FLOGI) or fabric discovery (FDISC).
- Brocade VDX hardware in the fabric maintains the MAC address, World Wide Name (WWN), and PID mappings per login. Each ENode port should have a unique MAC address and WWN.
- FIP FLOGI — The Brocade VDX hardware accepts the FIP FLOGI from the ENode. The FIP FLOGI acceptance (ACC) is sent to the ENode if the ENode MAC address or WWN matches the VN_Port table on the Brocade VDX hardware. The FIP FLOGI request is rejected if the ENode MAC address or WWN does not match. The ENode login is added to the VN_Port table. Fabric Provided MAC Addressing (FPMA) is supported.
- FIP FDISC — The Brocade VDX hardware accepts FIP FDISC from the ENode. FIP FDISC acceptance (ACC) is sent to the ENode if the ENode MAC address or WWN matches the VN_Port table on the Brocade VDX hardware. The FIP FDISC request is rejected if the ENode MAC address or WWN does not match. The ENode login is added to the VN_Port table. FPMA is supported.
- Maximum logins per VF_Port (logical FCoE port) — The Brocade VDX hardware supports a maximum of 64 logins. The VF_Port rejects further logins after the maximum is reached.
- Maximum logins per switch — The Brocade VDX hardware accepts a maximum of 1000 logins per switch.
- Maximum logins per VCS cluster — 3000.

FIP logout

FIP logout operates as follows:

- ENodes and VN_Ports can log out from the Brocade VDX hardware using FIP. The Brocade VDX hardware in the fabric updates the MAC address, WWN, and PID mappings upon logout. The Brocade VDX hardware also handles scenarios of implicit logout where the ENode has left the fabric without explicitly logging out.
- FIP logout (LOGO) — The Brocade VDX hardware accepts a FIP LOGO from the ENode. The FIP LOGO acceptance (ACC) should be sent to the ENode if the ENode MAC address and the VN_Port MAC address matches the VN_Port table data on the switch. The LOGO is ignored (not rejected) if the ENode MAC address does not match. The ENode logout is updated in the VN_Port table.

- Implicit logout — With the ENode directly connected to a DCB port, if the port that the ENode is attached to goes offline, the Brocade VDX hardware implicitly logs out that ENode. ENode logout is updated in the VN_Port table. The Brocade VDX hardware sends an FIP Clear Virtual Links (CVL) to the ENode.

  The FIP Virtual Link Maintenance protocols provide a mechanism to detect reachability loss to an Enode or any VN_Port instantiated on that ENode. This is accomplished by the periodic transmission of FIP Keep-Alive (FKA) messages from the ENode.

  If FKA timeouts are enabled on the switch, all VN_Ports associated with an ENode will be implicitly logged out in the event of an ENode FKA timeout.

  If FKA timeouts are enabled on the switch, the VN_Port will be implicitly logged out in the event of a VN_Port FKA timeout.

Name server operation

The Brocade VDX hardware name server function operates as follows:

- ENode login and logout to and from the Brocade VDX hardware updates the name server in the FC fabric. The Brocade VDX hardware maintains the MAC address to WWN and PID mappings.
- ENode login and logout — When an ENode login occurs through any means (FIP FLOGI, FIP FDISC, FCoE FLOGI, or FCoE FDISC), an entry is added to the name server. When an ENode logout occurs through any means (FIP LOGO, FCoE LOGO, or implicit logout), the entry is removed from the name server.
- ENode data — The Brocade VDX hardware maintains a VN_Port table. The table tracks the ENode MAC address, FIP login parameters for each login from the same ENode, and WWN and PID mappings on the FC side. You can display the VN_Port table with the **show fcoe login** command.

Registered State Change Notification

The Brocade VDX hardware Registered State Change Notification (RSCN) function operates as follows:

- RSCN events generated in the FC fabric are forwarded to the ENodes. RSCN events generated on the FCoE side are forwarded to the FC devices. DCB is not aware of RSCN events.
- Device RSCN — An RSCN is generated to all registered and affected members when an ENode either logs in or logs out of an FCF through any means. An RSCN is generated when an FC N_Port device either logs in or logs out of the FC fabric.

  NOTE
  When transmitting an RSCN, zoning rules still apply for FCoE devices as the devices are treated as regular FC N_Ports.

- VF_Port RSCN — An RSCN is generated to all registered members when a VF_Port goes online or offline, causing ENode or FC devices to be added or removed.
- Domain RSCN — An RSCN is generated to all registered and affected members when an FC switch port goes online or offline, causing ENode or FC devices to be added or removed. An RSCN is generated when two FC switches merge or segment, causing ENode or FC devices to be added or removed. When FC switches merge or segment, an RSCN is propagated to ENodes.
- Zoning RSCN — An RSCN is generated to all registered and affected members when a zoning exchange occurs in the FC fabric.

Local ENode configuration

The number of interfaces to be created is configured per switch by means of the **fcoe-enodes** command, executed in RBridge ID configuration mode. This is known as the local ENode configuration model. The number of ENodes that can be configured ranges from 0 through 1000, with a default of 64.

An FCoE license is required to enable FCoE interfaces. If this license is not present, no FCoE interfaces are created.

FCoE queuing

The QoS configuration controls the FCoE traffic distribution.

> NOTE
> Changing these settings requires changes on both the Brocade VDX hardware and the Converged Network Adapter (CNA); therefore, the link must be taken offline and put back online after a change is made.

Traffic scheduler configuration changes affect FCoE traffic distribution as follows:

- Changing the priority group for a port causes the FCoE traffic distribution to be updated. The priority group and bandwidth are updated.
- Changing the priority table for a port causes the FCoE traffic distribution to be updated. The CoS-to-priority group mapping is updated.
- Changing the class map for a port causes the FCoE traffic distribution to be updated.
- Changing the policy map for a port causes FCoE traffic distribution to be updated.
- Changing the DCB map for a port causes the FCoE traffic distribution to be updated.
- The FCMAP-to-VLAN mapping determines the FCoE VLAN allowed for the FCoE session. Modifying this mapping causes the existing sessions to terminate.

> NOTE
> Only one FCoE VLAN is supported in Network OS 4.0.0 and later releases.

FCoE logical SAN overview

The FCoE logical SAN feature supports up to four logical storage area networks in a VCS Fabric, in addition to the default FCoE VLAN SAN support. All nodes in a VCS Fabric must be upgraded to Network OS 6.0.0 or 6.0.1 for multiple FCoE logical SAN support..

A logical SAN provides Fibre Channel SAN connectivity to the FCoE device on any node in a VCS Fabric through an Access Gateway (AG) in the fabric or through a node with an E_Port configured to connect to a Brocade Fibre Channel Routing (FCR) EX_Port in a Fibre Channel SAN. Logical SANs can be configured as either local or remote SANs. A local logical SAN provides logical separation within the VCS Fabric, while a remote logical SAN provides FCoE SAN connectivity through an AG. Each AG can provide connectivity to only one remote logical SAN. Other switches in the VCS Fabric can be configured to support multiple logical SANs apart from the default SAN. Each logical SAN, identified by the fabric map configuration, uses a separate VLAN to provide traffic isolation.

For the local logical SANs within the VCS Fabric, the name server and Fabric Shortest Path First (FSPF) continue to treat all the SANs as a single fabric. On a local logical SAN, logins are serviced by the switch that is directly connected to the FCoE device.

For the default SAN, logins are serviced by the switch that is directly connected to the FCoE device. For the remote logical SANs, logins are serviced by the AG that is connected to the Fibre Channel SAN. The AG must be configured as an FCoE Forwarder (FCF) for that remote logical SAN. The non-AG switch, which is connected to the network by means of an FCoE converged network adapter (CNA) and acts only as a pass-through switch that sends traffic to the AG, is called an FCoE Initialization Protocol (FIP) forwarder, or FIF. The FIF-to-FCF mapping, whereby the FIF communicates with the FCF in the VCS Fabric, is determined by the FCF group configuration that they are part of. Intermediate switches that are connected between the FCF and the FIF need no special configuration.

The initial release of this feature (Network OS 6.0.0) supported four remote logical SANs within a single VCS Fabric, with each FIF supporting a default FCoE VLAN SAN and a remote logical SAN. Now, Beginning with Network OS 6.0.1, in addition to the default FCoE VLAN SAN, the local logical SAN feature is provided. In a VCS Fabric, the maximum logical SAN support is four logical SANs,

These four logical SANs can be all local. all remote, or a combinations of the two. Also starting Network OS 6.0.1, the FCoE port profile is enhanced to support logical SAN configuration.. it is now possible to map multiple local FCoE ports to a specified VLAN and fabric map, as well as to specified RBridge IDs, by means of the **fcport-group** command and the associated **fcport-group-rbid** command.

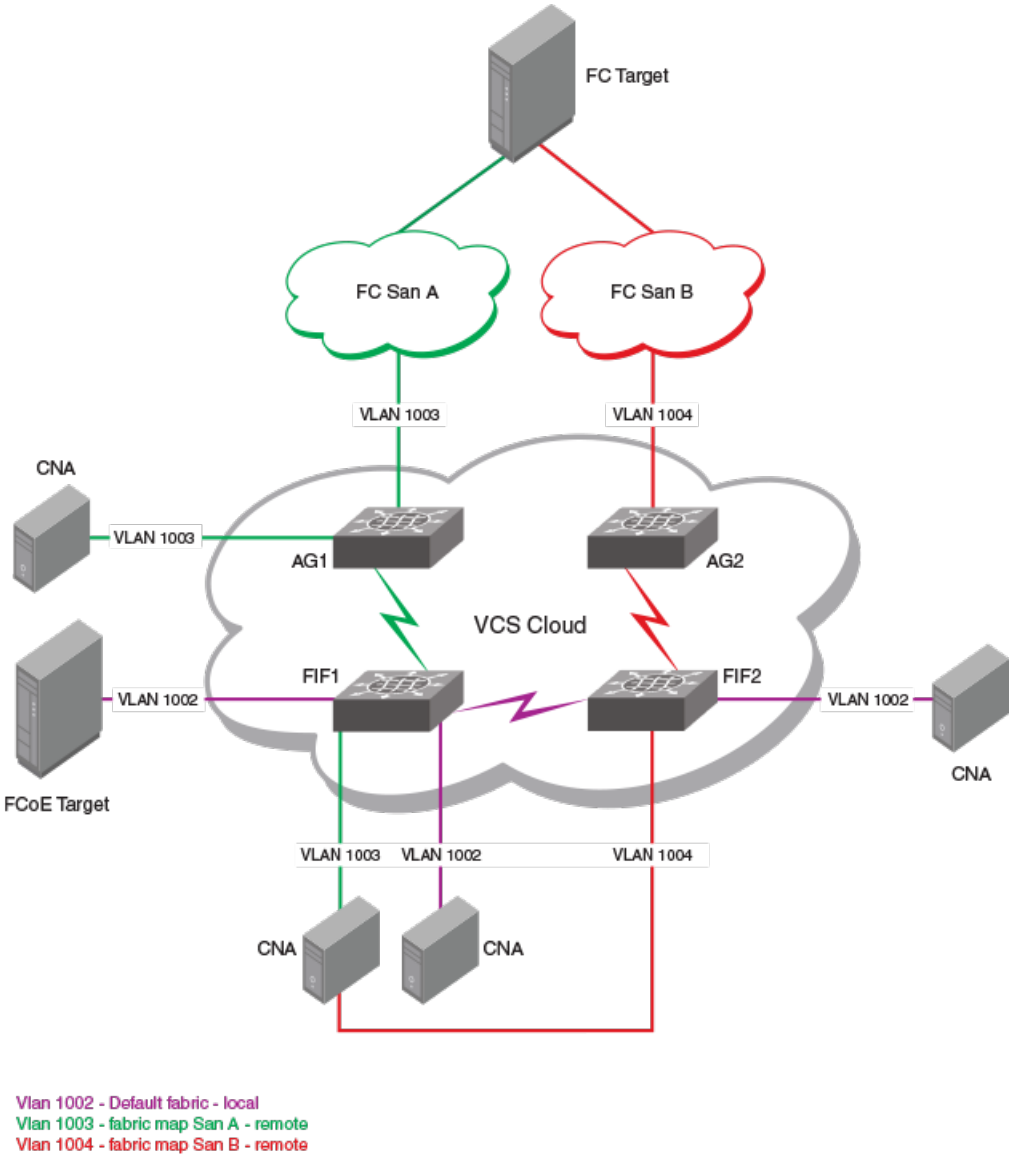
FCoE logical SAN use cases

The following topologies illustrate the three supported use cases.

FCoE logical SAN use case 1

In this use case, each FIF can support a default FCoE VLAN SAN and a remote logical SAN, as illustrated in the following figure. VLAN 1002 is local and supports the default fabric. VLANs 1003 and 1004 are remote and support SANs A and B, respectively.

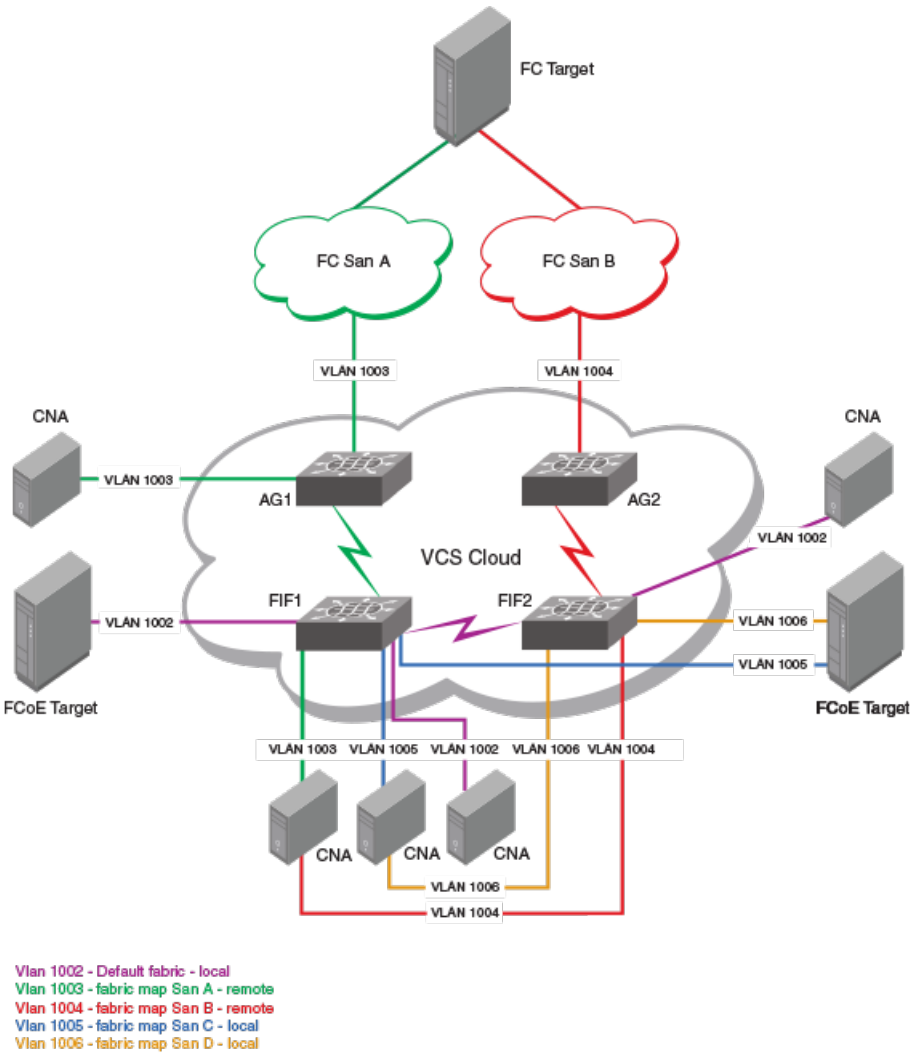
FIGURE 7 Use case 1: One remote SAN and a default FCoE VLAN SAN with a single FIF



FCoE logical SAN use case 2

In this use case, illustrated below, one remote logical SAN and one local logical SAN are allowed on a single FIF.

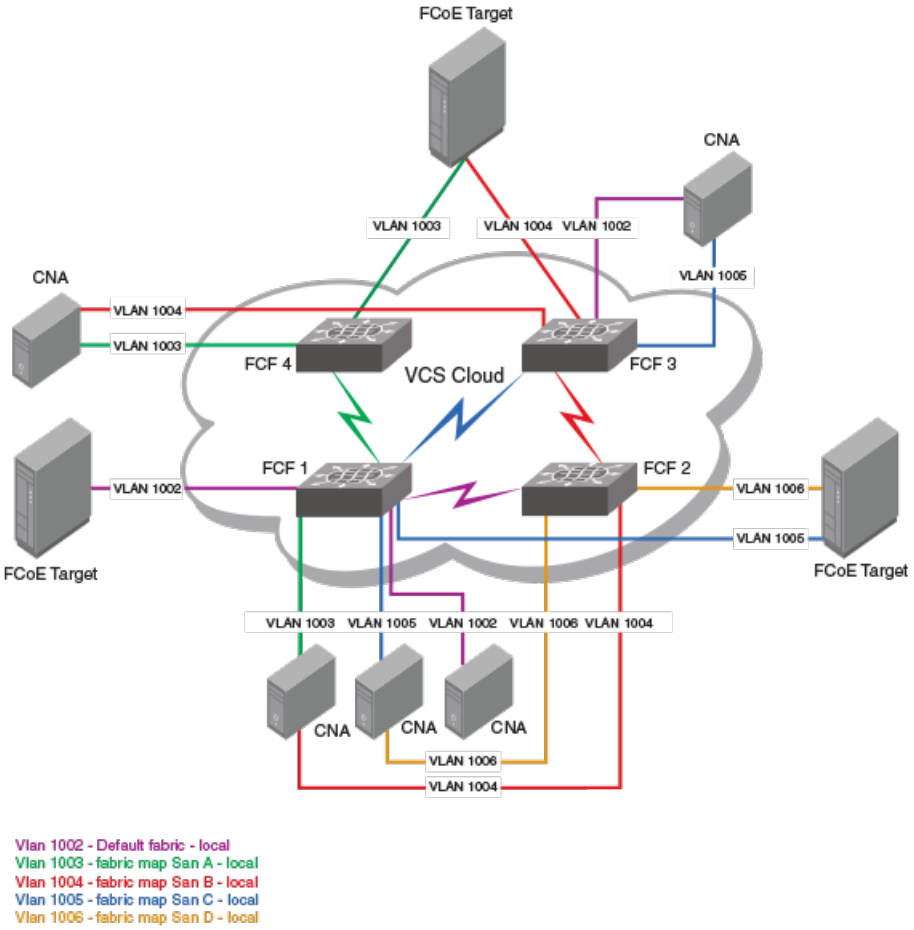
FIGURE 8 Use case 2: One remote logical SAN and one local logical SAN with a single FIF



FCoE logical SAN use case 3

In this use case, illustrated below, four logical SANs are supported. Each FIF supports two local logical SANs in addition to the default FCoE VLAN SAN. Local SANs do not connect to the FC SAN through the Access Gateway.

FIGURE 9 Use case 3: Four logical SANs



FCoE logical SAN behavior and provisioning model

When a VDX switch boots up in native FCoE mode (non-AG mode), its behavior is similar to the current local FCoE Forwarder (FCF) mode. The switch continues to support fcoeport configurations on the local FCoE VLAN as before, by means of the default configuration.

Logical SAN behavior and provisioning

The following summarizes the behavior and provisioning of FCoE logical SANs:

- Every logical SAN is represented by a fabric map, and has a unique FCoE VLAN ID and FCoE MAC address prefix (FCMAP). The fabric map must be identified as either "local" or "remote."
- Every remote logical SAN is represented by an FCF group. Each FCF group must have one RBridge as an FCF to service the remote logical SAN and can have multiple RBridges as FIFs to service the remote logical SAN.
- Local logical SANs are available for provisioning on all RBridges in the fabric. FC ports in an RBridge can be provisioned into a local logical SAN by means of the **fcport-group** command for the fabric map.

NOTE
FC ports provisioned in a local logical SAN can be in E_Port mode connecting to the EX end of an FC SAN through FCR, or can be in F_Port mode connecting to an FC end device.

- Ethernet ports can be provisioned for either local or remote logical SANs by means of the respective fabric maps for FCoE provisioning.

The following summarizes the configuration process:

- The user creates a fabric map by means of the **fabric-map** command, which in turn does the following automatically: **san-mode** is set to **remote**, one of the available VLANs is selected, and one of the available FCMAPs from the range 0E:FC:00 through 0E:FC:FF is selected. The user can change the VLAN and the FCMAP.
- Under the **fabric-map** command, the user uses the **fcf-group** command to create a group name and enter FCF group configuration mode. This mode allows the user to configure values for FCF and FIF RBridge IDs, set by the keywords **fcf-rbid** and **fif-rbid**, respectively. (A default fabric map does not contain an FCF group.)
- When the user attempts to configure an FCF RBridge, this is allowed only if the RBridge is enabled as an Access Gateway.
- In the case of fabric maps for local logical SANs, the user sets **san-mode** to **local** and uses the **fcport-group** command and the **fcport-group-id** subcommand to add RBridge IDs as appropriate. An "fcport-group" configuration can be part of a local fabric map only, and all FC ports in an attached RBridge are part of the local logical SAN.
- FCoE provisioning on interfaces is as before. The user uses the **fcoeport** command to configure physical interfaces for either local or remote logical SANs, as well as for the default FCoE logical SAN.
- The user applies FCoE provisioning on multiple ports by means of port profiles. The FCoE configuration is added to the FCoE subprofile of a port profile, and this profile is applied to ports as appropriate. (The FCoE subprofile configuration is applied like a VLAN subprofile and does not trigger MAC address learning on an interface.)
- The user associates each logical SAN to a port-profile domain, with each domain containing only one profile with the FCoE configuration. (Nondefault port profiles can have FCoE provisioning for only the nondefault fabric map.)
- Port-profile domains are configured only in Virtual Fabrics-enabled mode, and therefore port-profile support for logical SANs is provided only if Virtual Fabrics is enabled for all participating switches.

NOTE
FCoE provisioning for a logical SAN is not allowed on an Ethernet interface if the RBridge on which the interface is present is not part of the FCF group configuration, whether as an FCF RBridge ID or an FIF RBridge ID.

ATTENTION
The FCoE SAN configuration is global, and therefore must be done on the principal switch in logical chassis cluster mode. However, it is the user's responsibility to ensure that the configuration is the same on all participating switches and does not conflict with other switches in fabric cluster mode.

Understanding port profiles and multiple fabric maps

Although an FCoE configuration that is done as part of a profile is applied during the configuration and not as a result of MAC address learning, port-profile configuration allows the FCoE configuration to be aligned with the concept of domains. This provides a single point of configuration for all the ports that would form a connecting point in the VCS Fabric for a virtual machine (VM). (These are all ports to which a VM could be moved.) All ports that would support a particular VM must be configured with the fcoeport configuration for the appropriate SAN, so that the VM could boot from that FCoE SAN. This is achieved by mapping the logical SAN to one or more domains, so that each domain can have an FCoE configuration for only a particular SAN.
Prior to Network OS 6.0.0, FCoE profiles were supported only under the default port profile. Now the FCoE profile is supported under nondefault port profiles as well. Now the default port profile is provided only for backward compatibility, as well as for use cases in which the user wants to retain the default SAN and not use a logical SAN.

ATTENTION
It is recommended that only the domain-based application of port profiles (by means of the port-profile-port domain command) be used. A VCS Fabric can have port profiles with the default fcoe-map or the nondefault fcoe-map, but not both.

AMPP configurations

There are two Auto Migrating Port Profile (AMPP) configurations: global and interface level. These are discussed below with respect to the FCoE logical SANs feature.

Global port profiles

Global port profiles are configured by the **port-profile** command. The FCoE profile is configured by the **fcoeport** command, with the fabric map defined by the *fabric-map-name* variable. Nondefault maps are now supported. However, as these maps can be applied only through domains, Virtual Fabrics must be enabled to configure port profiles for nondefault SANs. As an FCoE port profile is applied during configuration and is not associated with any MAC address that must be learned, it is recommended that FCoE port profiles be kept separate from other port profiles.

The following is an example global port-profile configuration.

```
device(config)# port-profile A
device(config-port-profile-A)# fcoe-profile
device(config-fcoe-profile)# fcoeport sanA

device(config)# port-profile B
device(config-port-profile-B)# fcoe-profile
device(config-fcoe-profile)# fcoeport sanB

device(config)# port-profile C
device(config-port-profile-C)# fcoe-profile
device(config-fcoe-profile)# fcoeport sanA
```

NOTE
The default port profile can contain only the default FCoE map, and nondefault port profiles cannot contain the default FCoE map. Port-profile domains can contain many port profiles. However, only one port profile with an FCoE subprofile can be part of a domain. The same port profile can be part of multiple domains.

The following is an example global port-profile domain configuration.

```
device(config)# port-profile-domain D1
device(config-port-profile-domain-D1)# port-profile A
device(config-port-profile-domain-D1)# port-profile D

device(config)# port-profile-domain D2
device(config-port-profile-domain-D2)# port-profile A
device(config-port-profile-domain-D2)# port-profile E
```

Interface-level port profiles

Interface-level port profiles are configured by the **port-profile-port** command. The default FCoE port fabric map is automatically applied on an interface for default profiles where FCoE logical SANs are not enabled and the user wants to maintain a configuration created prior to Network OS6.0.1. In previous releases, the default port profile is applied on all port-profile ports, irrespective of the port-profile domain. This behavior continues unless nondefault port profiles are configured.

The fabric map configured by the **fcoeport** command is automatically applied on an interface where *fabric-map-name* is the fabric map under the profile with the FCoE subprofile that is part of the domain.

The following are example interface-level port-profile configurations.

Three logical SANs:

```
fabric-map sana
  fabric-map sanb
  fabric-map sanc
```

Three FCoE port profiles:

```
Port-profile ProfileSanA
        fcoeport sana
    Port-profile ProfileSanB
        fcoeport sanb
    Port-profile ProfileSanC
        fcoeport sanc
```

Five port-profile domains:

NOTE

These support various VMs identified by MAC address, and are applied when the associated MAC addresses are learned.

```
Port-profile-domain DD1
        Port-profile ProfileSanA
        Port-profile PP1
        Port-profile PP2
        Port-profile PP3

    Port-profile-domain DD2
        Port-profile ProfileSanB
        Port-profile PP1
        Port-profile PP4
        Port-profile PP5

    Port-profile-domain DD3
        Port-profile ProfileSanC
        Port-profile PP6
        Port-profile PP7

    Port-profile-domain DD4
        Port-profile ProfileSanA
        Port-profile PP5
        Port-profile PP3
        Port-profile PP7

    Port-profile-domain DD5
        Port-profile ProfileSanB
        Port-profile PP4
        Port-profile PP6
```

Restrictions for port profiles and fabric maps

Note the following restrictions. This behavior is enforced to ensure that configurations are consistent and undesired combinations of configurations are avoided.

- Default configurations are not allowed on nondefault port profiles.
- Only one port profile with an FCoE subprofile is allowed in a port-profile domain.
- Changing a fabric map or an FCF group is not allowed if the fabric map contains a port-profile-port configuration.
- FCoE configuration by means of the **fcoeport** and **port-profile-port** commands can coexist, but only if the FCoE map configurations that are inside the port-profile domain and also are applied directly on an interface are not in conflict.
- Changing an FCoE port profile is not allowed when either (a) the port profile is activated or (b) the FCoE map conflicts with the map applied on any interface.
- Adding a port profile to a port-profile domain is not allowed when either (a) the port-profile domain already has an FCoE map or (b) the port-profile domain is applied on at least one interface for which there is an conflicting FCoE map.

FCoE logical SAN limitations

Note the following limitations for this feature:

- One Access Gateway (AG) can service only one FCoE VLAN and one remote logical SAN.
- If the AG is not part of any remote logical SAN, the default logical SAN (fabric map) on the AG acts as a remote logical SAN. Therefore, it is not part of the default logical SAN (fabric map) on other Brocade VDX series platforms that are not AGs. The default logical SAN (fabric map) on non-AG VDX platforms acts as a local logical SAN, and all FCoE devices logged into the default local logical SAN on non-AG VDX platforms can see each other.
- An FIF at the first hop can be part of only one AG.
- An FIF or FCF can be part of only one FCF group.
- An FIF can support only two logical SANs (and a maximum of one remote logical SAN) in addition to the default.
- A maximum of four logical SANs (local, remote, or a combination of the two) are allowed in a single VCS Fabric.
- Each node in the VCS Fabric can support four local logical SANs.
- Fabric Services share the same device namespace between local logical SANs. Therefore, zoning needs to be used to isolate devices within a local logical SAN.
- FC ports (Flexports) in non-AG VDXs are by default part of the default logical SAN (fabric map). These ports can be made part of any other logical SAN by means of the **fcport-group** command; the fcport-group configuration can be part of only one local fabric-map. All FCoE ports on non-AG VDXs can be part of only one local logical SAN. Only RBridge IDs can be part of the fcport-group configuration, and all FC ports on a given RBridge are part of the logical SAN.
- FC ports in non-AG VDXs are part of the remote logical SAN that the AG is part of. If the AG is not made part of any remote logical SAN, those ports are part of the default logical SAN, which acts as a remote logical SAN.
- Duplicate WWN detection is done across all local logical SANs. Duplicate WWNs are not detected across local and remote or between two remote logical SANs in a VCS Fabric. The duplicate WWN detection for a remote logical SAN must be done at the upstream switch.

The following features and platforms are not supported:

- Pre-FIP versions of CNAs
- FIP version 0
- Hard zoning within a VCS Fabric
- FCoE in standalone mode (that is, VCS is disabled)
- FCoE over VLAG and Brocade LAG
- Traffic isolation (By default, the entire VCS Fabric network is treated as a single fabric. All logical SAN frames use the shorted VCS path available to reach the switch to which the frames are destined. Intermediate switches in the VCS Fabric are just pass-through devices and hence play the same role as they currently do.)
- The Brocade VDX 6730

For additional details related to this feature, see FCoE logical SAN behavior and provisioning model on page 52.

FCoE logical SAN upgrade/downgrade considerations

This section details the upgrade and downgrade considerations for this feature, including considerations for mixed-fabric deployments.

Upgrade considerations: NOS 5.0.x to NOS 6.0.x

- Non-ISSU upgrades from Network OS 5.0.x to Network OS 6.0.1 are supported.

- Following an upgrade, all existing FCoE configurations, including FCoE provisioning configurations on the Access Gateway, continue to be supported.
- The AG has the same functionality it had in Network OS 5.0.x.
- To use the new FCoE logical SAN features, the user must configure the required fabric maps and FCF groups for the logical SANs and accordingly change the FCoE interface provisioning on the AG.

Downgrade considerations: NOS 6.0.x to NOS 5.0.x

Non-ISSU downgrades from Network OS to Network OS 5.0.x are allowed only when there are no FCoE logical SAN configurations.

Upgrade considerations: NOS 6.0.0 to NOS 6.0.1

- All FCoE login sessions are restored after the upgrade. Devices re-login because the system goes through a cold recovery.
- To use the local logical SAN feature, the provisioning model must be followed to migrate from the default/remote FCoE VLAN to a new local FCoE VLAN, and configurations must be changed as appropriate.

Downgrade considerations: NOS 6.0.1 to NOS 6.0.0

- If the switch is configured with the local logical SAN feature, a downgrade is prevented.
- If FCoE is configured on any nondefault port profiles, the downgrade is prevented.
- 
  CAUTION
  A downgrade is disruptive.

FCoE logical SAN scalability

The default limit is 64 ENodes. The following table lists the maximum limits for this feature.

TABLE 4 Maximum limits for FCoE logical SANs

Parameter	Limit
Virtual Fabrics (VF) ports per switch	1000
N_Port ID Virtualization (NPIV) instances per port	64
FCoE devices per VCS Fabric (FLOGI + FDISC)	2000
Total SAN devices per VCS Fabric (including fabric cluster)	3000
LLDP neighbors per switch	VDX 8770: 384 VDX 67xx: 60 VDX 6940: 112
Total number of logical SANs	4, plus 1 default SAN
Number of default Enodes	64

FCoE logical SAN configuration recommendations

Note the following guidelines, which apply to changing both default and nondefault parameters:

- Configuration changes are not allowed when logins are present.
- Interfaces cannot be provisioned for FCoE (especially those for logical SANs) if the entire configuration for a logical SAN is not in place. All fabric map and FCF group configurations must be complete.

- The deletion of any of the following logical SAN components is not allowed when there are ports provisioned for a logical SAN: the fabric map or FCF group, the FCF RBridge ID, or the FIF RBridge ID.

Configuring FCoE

This section presents the tasks necessary to configure FCoE interfaces and logical SANs.

Configuring logical FCoE ports

When the switch boots, a pool of 64 FCoE ports is created. These ports are not bound to any physical ports. The bindings are created when an FLOGI is received on the switch. Any free port that is available from the pool is selected and bound to the physical port where the FLOGI is received. The default number of logical ports is 64, and the range of valid values is from 0 through 1000.

NOTE
Brocade VDX switches support FCoE multi-hops for as many as nine hops.

When the FCoE logical port is automatically bound to a 10-gigabit Ethernet LAG port, this is referred to as *dynamic binding*. This binding is valid only until the FLOGI session is valid. The binding is automatically removed when CNA logs out. To create a persistent binding between the logical FCoE port and an interface that can be used for static binding (FortyGigabitEthernet, HundredGigabitEthernet, Port-channel, TenGigabitEthernet, mac-address), use the **bind** command. This is stored in the configuration and retained across reboots.

NOTE
Only one type of binding can be used for each physical port, so the LAG binding configurations will overwrite each other.

To create additional logical FCoE ports, perform the following steps in RBridge ID configuration mode.

1. Enter FCoE configuration mode on an RBridge.

```
device(config-rbridge-id-1)# fcoe
device(config-rbridge-fcoe)#
```

2. Enter the **fcoe-enodes** command to set the maximum number of logins allowed on the switch.

```
device(config-rbridge-fcoe)# fcoe-enodes 384
```

To bind the logical FCoE ports to a physical port, perform the following steps:

3. Enter interface subtype configuration mode on the RBridge.

```
device(config-rbridge-id-1)# interface fcoe 1/1/55
device(config-if-fcoe-1/1/55)#
```

4. Bind the logical port to the physical port.

```
device(conf-if-fcoe-1/1/55)# bind tengigabitethernet 1/0/1
```

5. In privileged EXEC mode, verify the FCoE ENode configuration, by using the **show fcoe fcoe-enodes** command.

```
device# show fcoe fcoe-enodes
================================================
Rbridge-id           Fcoe-enodes
================================================
1                       384
2                       64[D]
6                    0
```

Configuring fabric maps

Fabric maps are used to configure FCoE properties on interfaces.

> NOTE
> This is not supported for remote or local FCoE logical SANs.

A fabric map is a placeholder for an FCoE VLAN and an FCMAP.

A fabric map with the name "default" is created during system boot-up. The user is not allowed to delete or rename this map. By default, the FCoE VLAN associated with the fabric map is the native FCoE VLAN 1002, the 802.1Q priority associated with the fabric map is 3. and the associated FCMAP is 0E:FC:00. The user can modify the VLAN, the priority, or the FCMAP, but cannot delete any of these.

> NOTE
> A fabric map can be edited even if it is associated with an interface. However, the VLAN of the fabric map cannot be edited. All other parameters, such as the FCMAP, priority, and advertisement interval, can be modified.

Do the following to edit the parameters of the default fabric map.

1. In global configuration mode, enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

2. Enable the default fabric map and modify it as in the following example.

```
device(config-fcoe)# fabric-map default
device(config-fcoe-fabric-map-default)# vlan 1003
device(config-fcoe-fabric-map-default)# priority 4
device(config-fcoe-fabric-map-default)# fcmap 0e:fc:11
```

> NOTE
> Brocade does not support non-FCoE traffic over an FCoE VLAN. The FCoE VLAN should not carry any mixed traffic.

Configuring FCoE logical SANs

This section presents the tasks required to configure FCoE logical SANs and manage those configurations. Refer to FCoE logical SAN use case 2 on page 50 for this example.

The VLANs are assigned as follows (if they have not already been assigned).

> NOTE
> The first free unprovisioned VLANs are assigned to the fabric map.

VLAN	Fabric	Logical SAN (san-mode)
1002	Default	Local
1003	fabric map SanA	Remote
1004	fabric map SanB	Remote
1005	fabric map SanC	Local
1006	fabric map SanD	Local

The configuration is global and consists of the following:

1. Configuring the FCoE logical SAN fabric maps to identify the nondefault SANs

2. For remote logical SANs, configuring the FCoE logical SAN FCF groups to map the FIFs to FCFs within a fabric map of nondefault SANs
3. Configuring the Access Gateways as FCoE Forwarder (FCF) RBridge IDs and their participation in the nondefault SANs

NOTE
It is also possible to map multiple local FCoE ports to a specified VLAN and fabric map, as well as to specified RBridge IDs,

For the configurations of the remaining use cases, refer to FCoE logical SANs configuration examples on page 67.

Creating fabric maps for logical SANs

This task creates fabric maps for a nondefault SAN.

Do the following to create and configure a fabric map for nondefault SAN SanA.

1. In global configuration mode, enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

2. Create the fabric map instance and enter FCoE fabric-map configuration mode.

```
device(config-fcoe)# fabric-map SanA
device(config-fcoe-fabric-map-SanA)#
```

NOTE
Values for VLAN, priority, and FCMAP are selected automatically from available values.

3. Change the VLAN from the default.

```
device(config-fcoe-fabric-map-SanA)# vlan 1003 [was 1004]
```

4. Change the priority from the default.

```
device(config-fcoe-fabric-map-SanA)# priority 3 [was 4]
```

NOTE
The priority should be the same for all fabric maps.

5. Change the fabric-provided MAC address (FPMA) FCoE MAC address prefix (FCMAP) from the default.

```
device(config-fcoe-fabric-map-SanA)# fcmap 0e:fc:10 [was 0e:fc:01]
```

NOTE
With multiple fabric maps, each has its own FCMAP value. Values must be unique across all fabric maps. The **no fcmap** command does not allow reversion to the default FCMAP value for a particular fabric map.

6. (Optional) Change the advertisement (FCoE keep-alive, or FKA) interval from the default of 8000.

```
device(config-fcoe-fabric-map-SanA)# advertisement interval 10000
```

7. Repeat the above for SanB, SanC, and SanD with appropriate values.

Use the **show fcoe fabric-map** command to confirm the current status of the FCoE fabric map, as in the following example.

```
device# show fcoe fabric-map SanA
=================================================================
Fabric-Map VLAN VFID   Pri FCMAP    FKA   Timeout
=================================================================
SanA       1004 128[D] 4   0x0efc01 10000 Enabled[D]
Total number of Fabric Maps = 1
```

Configuring the FCF groups for FCoE logical SANs

The Access Gateways (AGs) must be configured to identify the membership of the FCoE Forwarder (FCF), as well as the RBridge IDs of the forwarding RBridges. Those RBridges must be first-hop switches. All intermediate switches forward the SAN traffic by default.

Do the following to configure the AGs to support FCoE logical SANs. Refer to FCoE logical SAN use case 1 on page 49. Once the membership is configured, you must configure FCF groups and fabric maps for the nondefault SANs.

1. In global configuration mode, enter the **fcoe** command to enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

2. Enter the **fabric-map** command and specify a nondefault SAN, entering FCoE fabric-map configuration mode.

```
device(config-fcoe)# fabric-map SanA
device(config-fcoe-fabric-map-SanA)#
```

3. In FCoE fabric-map configuration mode, specify an FCF group to enter FCoE FCF group configuration mode, and specify an FCF RBridge ID and one or more FIF RBridge IDSs.

   NOTE
   Use the **add** keyword to add multiple RBridge IDs. Alternatively, use the **remove** keyword to remove RBridge IDs. Ranging (with a hyphen) is also allowed.

```
device(config-fcoe-fabric-map-SanA)# fcf-group rack-1
sw0(config-fabric-map-fcf-group-rack-1)# fcf-rbid 6
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid add 5
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid add 10-15
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid add 28-30,35
sw0(config-fabric-map-fcf-group-rack-1)# fif-rbid remove 28-30,12
```

4. You can verify the FCF map configuration by viewing the running configuration, or by entering the **show fcoe fcf-group** command as in the following examples:

```
device(config-fabric-map-fcf-group-rack-2)# do show running-config fcoe fabric-map fcf-group

fcoe
 fabric-map sanA
  fcf-group rack-1
   fcf-rbid  100
   fif-rbid add 5,10-11,13-15,35
  !
  fcf-group rack-2
    fcf-rbid 150
   fif-rbid add 180
  !
 !
 fabric-map sanB
  fcf-group rack-3
   fcf-rbid  200
   fif-rbid add 220, 221

device(config-fabric-map-fcf-group-rack-2)# do show fcoe fcf-group
==========================================================================
FCF-Group    Fabric-Map  FCF_RBID               FIF_RBID(s)
==========================================================================
rack-1     sanA         100             5   10  11  13  14  15
                                        35
rack-2     sanA         150             180
rack-3     sanB         200             220, 221

Total number of FCF Groups = 3
```

Configuring FCoE logical SAN port groups

This task maps one or more local FCoE ports to a specified VLAN, enabling the addition or removal of FCoE Initialization Protocol (FIP) Forwarder (FIF) RBridge IDs to or from an FCoE Forwarder (FCF) group.

Currently only one local FCoE VLAN is supported. In a VCS Fabric cluster with multiple hosts and targets connected to switches by means of Flex ports, only the default FCoE VLAN (1002) can be used. Therefore, to map multiple local FC ports a particular (nondefault) VLAN, use the **fcport-group** and **fcport-group-rbid** commands. Note the following conditions:

- An FC port group is the same as an FCF group under a fabric map.
- An FC port group can only be created when the fabric map san-mode is "local".
- The FC port group has a 1:1 relationship with the fabric map.
- When the fabric map is deleted, the FC port group is deleted automatically.
- The **fcport-group-rbid** command, under the **fcport-group** command, is used to add and remove RBridge IDs.

Do the following to enable the addition or removal of FCoE Initialization Protocol (FIP) Forwarder (FIF) RBridge IDs to or from an FCoE Forwarder (FCF) group.

1. In global configuration mode, enter FCoE configuration mode.

```
device(config)# fcoe
device(config-fcoe)#
```

2. Enter the fabric-map command and specify a logical SAN fabric map.

```
device(config-fcoe)# fabric-map SanA
device(config-fcoe-fabric-map-SanA)#
```

3. Enter the **fcport-group** command, to enter FCoE port-group configuration mode, and then enter the **fcport-group-rbid** command to specify one or more RBridge IDs.

```
device(config-fabric-map-fcport-group-SanA)# ?
Possible completions:
  describe            Display transparent command information
  do                  Run an operational-mode command
  exit                Exit from current mode
  fcport-group-rbid   Configure an FCPORT group rbridge-ids.
  help                Provide help information
  no                  Negate a command or set its defaults
  pwd                 Display current mode path
  top                 Exit to top level and optionally run command
```

4. In FCoE port-group configuration mode, specify an RBridge ID.

```
device(config-fabric-map-fcport-group-SanA)# fcport-group-rbid add 29
```

NOTE

Ranging and comma delimiters are allowed for multiple RBridge ID entries. Use the **remove** keyword to remove one or more RBridge IDs.

5. To confirm the configuration, use the **show runnng-config-fcoe** command.

```
device(config-fabric-map-fcport-group-SanA)# do show running-config fcoe
fcoe
fabric-map default
  vlan 1002
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0E:FC:00
  advertisement interval 8000
  keep-alive timeout
!
```

```
fabric-map SanA
  vlan 4
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0E:FC:03
  advertisement interval 8000
  keep-alive timeout
  fcport-group
   fcport-group-rbid add 29
  !
```

Assigning a fabric map onto an interface

The user assigns a fabric map onto an Ethernet interface by using the **fcoeport** command.

This configuration is called *FCoE provisioning*. Once the fabric map is assigned onto an interface, the following are applied to the interface:

- The corresponding FCoE VLAN
- The default CEE map
- The FCoE/FIP VLAN classifiers

In short, the interface becomes capable of carrying FCoE traffic.

> NOTE
> The fabric map can be applied irrespective of whether or not the interface is in "switchport" mode. However, the fabric map cannot be applied on an interface if the same interface already has a CEE map assigned to it.

Do the following to assign a fabric map onto an interface.

1. Activate interface subtype configuration mode for the interface to be modified.

```
device(config)# interface tengigabitethernet 1/0/1
device(conf-if-te-1/0/1)#
```

2. Apply the current fabric map to the interface by using the **fcoeport** command and entering "default" as the map name.

```
device(conf-if-te-1/0/1)# fcoeport default
```

> NOTE
> The default configuration continues to be supported. However, beginning with Network OS 6.0.0, user-specified map names are allowed to support FCoE logical SANs.

3. Return to privileged EXEC mode by using the **end** command.

```
device(conf-if-te-1/0/1)# end
```

4. Confirm the changes to the interface by using the **show running-config** command.

```
device# show running-config interface tengigabitethernet 1/0/1
interface TenGigabitEthernet 1/0/1
fcoeport default
no shutdown
```

5. Use the **fcoe fabric-map default** command to confirm the current status of the fabric map.

```
device# show fcoe fabric-map
=====================================================================
Fabric-Map VLAN    VFID    Pri  FCMAP      FKA     Timeout
=====================================================================
default    1002[D] 128[D] 3[D] 0xefc00[D] 8000[D] Enabled[D]
Total number of Fabric Maps = 1
```

6. Repeat this procedure for any additional interfaces as appropriate.

Assigning an FCoE fabric map onto a LAG member

The **fcoeport** command is used under interface subtype configuration mode to provision a port to be an FCoE port. This puts the port in Layer 2 mode, but only for FCoE VLANs. In Network OS 4.0.0 and later, the **fcoeport default** command is supported for LAG member ports where FCoE provisioning is applied to individual Ethernet ports.

For all LAGs with FSB, the **fcoeport config** command must be applied on the LAG itself. For all LAGs with directly attached CNAs, the **fcoeport config** command must be applied on the member ports. Once this command is applied, and if the member port of the LAG is CEE-capable, the port carries FCoE traffic only.

NOTE
Note the following conditions:

- FCoE provisioning is allowed on a LAG member only if the LAG is not FCoE provisioned.
- The default configuration continues to be supported. However, beginning with Network OS 6.0.0, user-specified fabric maps are allowed to support multiple FCoE logical SANs.

To assign the FCoE fabric map onto a LAG member, perform the following steps.

1. Activate interface subtype configuration mode for the interface to be modified.

```
device(config)# interface tengigabitethernet 3/0/19
switch(conf-if-te-3/0/19)#
```

2. Activate the channel-group mode.

```
device(conf-if-te-3/0/19)# channel-group 10 mode active type standard
```

3. Set the LACP timeout to **long**.

```
device(conf-if-te-3/0/19)# lacp timeout long
```

4. Apply the current FCoE fabric map to the interface by using the **fcoeport default** command.

```
device(conf-if-te-3/0/19)# fcoeport default
```

5. Return to privileged EXEC mode by using the **end** command.

```
device(conf-if-te-3/0/19)# end
```

6. Confirm the changes to the interface with the **show running-config** command.

```
device# show running-config interface tengigabitethernet 3/0/19

interface TenGigabitEthernet 3/0/19
    fabric isl enable
    fabric trunk enable
    channel-group 10 mode active type standard
    lacp timeout long
    fcoeport default
    no shutdown
```

7. Use the **show fcoe interface brief** command to confirm the current status of the FCoE logins.

```
device# show fcoe interface brief
```

8. Repeat this procedure for additional interfaces as appropriate.

Configuring FCoE over LAG

Network OS 4.0.0 and later supports FCoE over LAGs. These are LAGs between the FCoE Forwarder (FCF) and a DCB-capable switch. The entire LAG is provisioned for FCoE, so that all member ports are used for FCoE traffic. FCoE traffic is broadcast on all the member links of the LAG.

> NOTE
> FCoE over LAG supports standard LAGs only; vLAGs are not supported.

Additionally, Network OS 4.0.0 and later supports multiple logins per port. This feature allows multiple ENodes to log in to a single 10-gigabit Ethernet port or a LAG.

Guidelines and restrictions for configuring FCoE over LAG

Follow these configuration guidelines and restrictions when configuring FCoE over LAG:

- The intermediate switches may or may not be an FSB. However, FSB is recommended for security.
- All ACLs and FCoE forwarding entries will continue to be on the FCF's ingress ports.
- It is assumed that the intermediate switch works in "Willing" mode towards the FCF in the DCBX exchange, and accepts the configuration from the FCF and propagates it downstream.
- The CEE/DCBX configuration is expected to identical on both the FCF and the intermediate switch.
- Irrespective of the previous two items, the PFC/No-drop behavior from the FCF's perspective will be guaranteed only on the links between the FCF and the first-hop switch. There is no provision in the standard to guarantee this requirement on all paths leading to the Enode.
- FSBs may or may not be able to forward the FCoE LLS TLV to the Enodes. Hence this TLV may not be present in the LLDP packets sent to the Enodes. The FCF continues to send this TLV in its LLDP packets destined to the intermediate switch.
- The default map configuration continues to be supported on LAG port-channels, as shown in the following section. However, beginning with Network OS 6.0.0, user-specified fabric maps are allowed to support multiple FCoE logical SANs.

Configuring FCoE provisioning on LAGs

The **fcoeport** command has been extended to the LAG interfaces to support the logical SANs feature, as shown in the following example.

```
switch# configure
Entering configuration mode terminal

switch(config)# interface port-channel 10

switch(config-Port-channel-10)# fcoeport default

switch(config-Port-channel-10)#
```

This provisions all the member ports of port-channel 10 for FCoE.

> NOTE
> The default configuration continues to be supported. (The keyword "default" must be entered manually.) However, beginning with Network OS 6.0.0, user-specified fabric maps are allowed to support multiple FCoE logical SANs.

Configuring interfaces to support FCoE logical SANs

To assign the logical FCoE fabric maps onto an interface, use the logical SAN fabric map instead of the "default" map as the map name in the **fcoeport** command. The guidelines for provisioning FCoE on physical interfaces, LAG members, and LAGs is the same as for the default case.

The following example provisions the 10-gigabit Ethernet interface in slot 0, port 1 on RBridge ID 1.

1. Activate interface configuration mode for the interface to be modified.

   The following example activates the mode for the 10-gigabit Ethernet interface in slot 0/port 1 on RBridge ID 1.

   ```
   switch(config)# interface tengigabitethernet 1/0/1
   switch(conf-if-te-1/0/1)#
   ```

2. Apply the current FCoE fabric map to the interface by using the **fcoeport** command with the desired map name.

   ```
   switch(conf-if-te-1/0/1)# fcoeport SanA
   ```

3. Return to privileged EXEC mode by using the **end** command.

   ```
   switch(conf-if-te-1/0/1)# end
   switch#
   ```

4. Confirm the changes to the interface by using the **show running-config** command.

   ```
   switch# show running-config interface tengigabitethernet 1/0/1
   interface TenGigabitEthernet 1/0/1
    fcoeport SanA
    no shutdown
   ```

5. Repeat this procedure for additional interfaces as appropriate.

Verifying FCoE logical SAN configurations

The following table lists **show** commands that can be used to verify FCoE logical SAN configurations. For details, refer to the *Network OS Command Reference*.

TABLE 5 Commands to verify FCoE logical SAN configurations

Command	Description
show fcoe devices	Displays the FCoE devices information.
show fcoe fabric-map	Displays the FCoE fabric-map configuration globally in a fabric, or on a single RBridge.
show fcoe fcf-group	Displays FCF groups information.
show fcoe fcoe-enodes	Displays FCoE ENodes information.
show fcoe fcoe-map	Displays information about the FCoE map.
show fcoe interface ethernet	Displays a synopsis of the FCoE Ethernet interfaces.
show fcoe fcport-group	Displays RBridge IDs and FCoE port connector groups associated with a fabric map.
show fcoe login	Displays FCoE login information.

FCoE logical SANs configuration examples

This section summarizes the configurations for use cases 2 and 3.

Use case 2 configuration example

The VLANs are assigned as follows.

VLAN	Fabric	Logical SAN
1002	Default	Local
1003	fabric map sanA	Remote
1004	fabric map sanB	Remote
1005	fabric map sanC	Local
1006	fabric map sanD	Local

```
fcoe
 fabric-map default
  vlan 1002
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0E:FC:00
  advertisement interval 8000
  keep-alive timeout
 !

 fabric-map sanA
  vlan 1003
  san-mode remote
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:10
  advertisement interval 8000
  keep-alive timeout
  fcf-group sanA
   fcf-rbid 1
   fif-rbid add 2
  !
!

 fabric-map sanB
  vlan 1004
  san-mode remote
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:20
  advertisement interval 8000
  keep-alive timeout
  fcf-group sanB
   fcf-rbid 3
    fif-rbid add 4
   !
 !

 fabric-map sanC
  vlan 1005
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:30
  advertisement interval 8000
  keep-alive timeout
 !

 fabric-map sanD
  vlan 1006
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:40
```

```
 advertisement interval 8000
 keep-alive timeout
```

Use case 3 configuration example

The VLANs are assigned as follows.

VLAN	Fabric	Logical SAN
1002	Default	Local
1003	fabric map sanA	Local
1004	fabric map sanB	Local
1005	fabric map sanC	Local
1006	fabric map sanD	Local

```
fcoe
 fabric-map default
  vlan 1002
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0E:FC:00
  advertisement interval 8000
  keep-alive timeout
 !

 fabric-map sanA
  vlan 1003
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:10
  advertisement interval 8000
  keep-alive timeout
 !

 fabric-map sanB
  vlan 1004
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:20
  advertisement interval 8000
  keep-alive timeout
 !

 fabric-map sanC
  vlan 1005
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:30
  advertisement interval 8000
  keep-alive timeout
 !

 fabric-map sanD
  vlan 1006
  san-mode local
  priority 3
  virtual-fabric 128
  fcmap 0e:fc:40
  advertisement interval 8000
  keep-alive timeout
 !
```

Managing duplicate WWNs

This feature enables the management of conflicts resulting from duplicate port WWNs arising from multiple causes.

It is never appropriate for two devices with the same port WWN (WWPN) to be in the same Name Server at the same time, as this results in incorrect responses to commands and unpredictable results, for a variety of reasons:

- *Timing issues*: Because Fibre Channel data bases are distributed, one device can appear to be logged in through two unique port locations.
- *Notification errors*: The removal of a device is not recognized and the device logs back in with a different port address, as if there were two devices.
- *Actual duplication*: Two distinct devices with the same WWPN attempt to log into the fabric at the same time. (This issue is not resolvable without authenticating the devices or using access control to limit which devices are allowed to log into specific ports.)

Beginning with Network OS 6.0.1, the user can specify system behavior when duplicate WWPNs are detected, by using the **fabric login-policy** command on a specified RBridge. This is a node-specific cluster command whose effect is persistent across reboots. A secondary node must be configured only through the principal node in management cluster or logical chassis cluster modes. When duplicate WWPNs are detected across two switches, the Name Server posts RASLOG NS-1012.

NOTE
This feature is not available on a device that is configured for Access Gateway mode.

The keywords below enable the following options:

Keyword	Description
new-login	Allows the "new" device to log in and cleans up the "old" (previous) login.
old-login	Allows the "old" device to retain the login and rejects the "new" login. This is the default.

NOTE
For reasons noted above, the concepts "old" and "new" are relative and not necessarily literal. The methods required to correct actual duplicate WWPN conditions within a fabric are beyond the scope of this feature.

To view the current configuration of this feature, use the **show fabric login-policy** command to view the login policy for a local node, a specific RBridge, or all RBridges in the fabric.

802.1Q VLANs

802.1Q VLAN overview

NOTE

This chapter addresses the use of standard Virtual LANs (VLANs) as defined by IEEE 802.1Q. Those VLANs have VLAN IDs that range from 1 through 4096. To support multitenancy by means of classified VLANs, the ID range has been extended through 8191. For details on this feature, refer to Virtual Fabrics on page 103.

IEEE 802.1Q VLANs provide the capability to overlay the physical network with multiple virtual networks. VLANs allow you to isolate network traffic between virtual networks and reduce the size of administrative and broadcast domains.

A VLAN contains end stations that have a common set of requirements that are independent of physical location. You can group end stations in a VLAN even if they are not physically located in the same LAN segment. VLANs are typically associated with IP subnetworks and all the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. VLAN membership is configurable on a per-interface basis.

The VLAN used for carrying FCoE traffic needs to be explicitly designated as the FCoE VLAN. FCoE VLANs are configured through the Network OS CLI (refer to Configuring an interface port as a Layer 2 switch port on page 75 for details).

NOTE

Currently only one VLAN can be configured as the FCoE VLAN at a time.

Ingress VLAN filtering

A frame arriving at Brocade VDX hardware is either associated with a specific port or with a VLAN, based on whether the frame is tagged or untagged. The association rules are as follows:

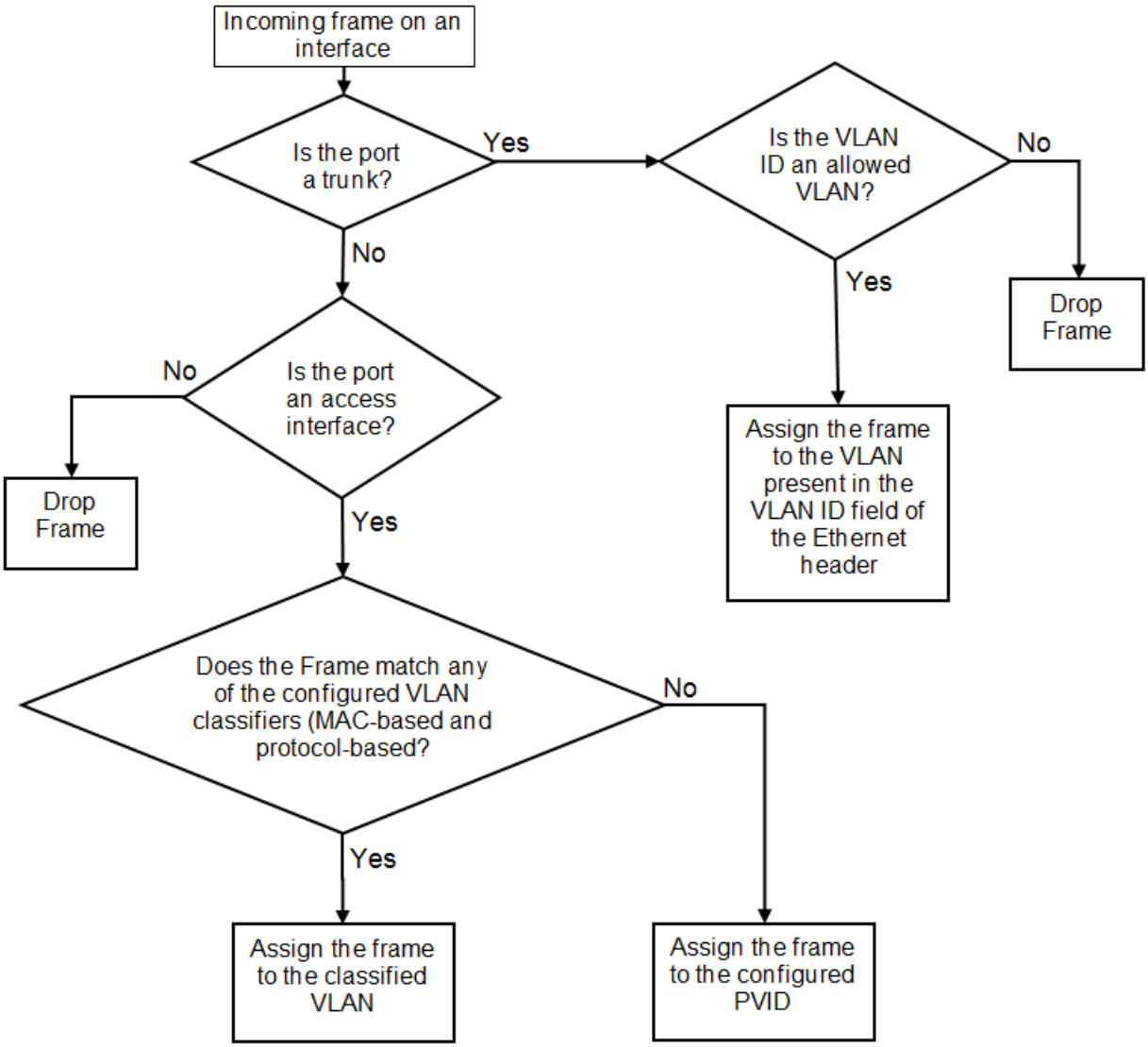
- Admit tagged frames only — The port the frame came in on is assigned to a single VLAN or to multiple VLANs depending on the VLAN ID in the frame's VLAN tag. This is called trunk mode.
- Admit untagged frames only — These frames are assigned the port VLAN ID (PVID) assigned to the port the frame came in on. This is called access mode.
- Admit VLAN tagged and untagged frames — All tagged and untagged frames are processed as follows:
  - All untagged frames are classified into native VLANs.
  - If the tengigabitethernet interface port is configured as an fcoeport and is in access mode, untagged Layer 2 or priority-tagged frames are forwarded by the egress port as untagged frames, unless you enable priority-tagging on the tengigabitethernet interface. By default, priority-tagging is disabled.
  - Any tagged frames coming with a VLAN tag equal to the configured native VLAN are processed.
  - For ingress and egress, non-native VLAN tagged frames are processed according to the allowed VLAN user specifications. This is called trunk mode.

NOTE

Ingress VLAN filtering is enabled by default on all Layer 2 interfaces. This ensures that VLANs are filtered on the incoming port (depending on the user configuration).

The following illustrates the frame-processing logic for an incoming frame.

FIGURE 10 Ingress VLAN filtering



There are important facts you should know about Ingress VLAN filtering:

- Ingress VLAN filtering is based on port VLAN membership.
- Port VLAN membership is configured through the Network OS CLI.
- Dynamic VLAN registration is not supported.
- The Brocade VDX hardware does VLAN filtering at both the ingress and egress ports.
- The VLAN filtering behavior on logical Layer 2 interfaces such as LAG interfaces is the same as on port interfaces.
- The VLAN filtering database (FDB) determines the forwarding of an incoming frame.

Additionally, there are important facts you should know about the VLAN FDB:

- The VLAN FDB contains information that helps determine the forwarding of an arriving frame based on MAC address and VLAN ID data. The FDB contains both statically configured data and dynamic data that is learned by the switch.
- The dynamic updating of FDB entries using learning is supported (if the port state permits).
- Dynamic FDB entries are not created for multicast group addresses.
- Dynamic FDB entries are aged out based on the aging time configured per Brocade VDX hardware. The aging time is between 60 and 1000000 seconds. The default is 300 seconds.
- You can add static MAC address entries specifying a VLAN ID. Static entries are not aged out.
- A static FDB entry overwrites an existing dynamically learned FDB entry and disables learning of the entry going forward.

NOTE
For more information on frame handling for Brocade VDX hardware, refer to FCoE and Layer 2 Ethernet on page 40.

VLAN configuration guidelines and restrictions

Follow these guidelines and restrictions when configuring VLANs:

- On all Brocade VDX switches, VLAN 1002 is reserved for FCoE VLAN functionality.
- On Brocade VDX 8770 switches: 1 through 4086 for 802.1Q VLANs (VLAN IDs 4087 through 4095 are reserved on these switches), and 4096 through 8191 for service or transport VFs in a Virtual Fabrics context.
- On all other Brocade VDX switches: 1 through 3962 for 802.1Q VLANs (VLAN IDs 3963 through 4095 are reserved on these switches), and 4096 through 8191 for service or transport VFs in a Virtual Fabrics context.
- In an active topology, MAC addresses can be learned, per VLAN, using Independent VLAN Learning (IVL) only.
- A MAC address ACL always overrides a static MAC address entry. In this case, the MAC address is the forwarding address and the forwarding entry can be overwritten by the ACL.
- The Brocade DCB switch supports Ethernet DIX frames and 802.2 LLC SNAP encapsulated frames only.
- You must configure the same native VLAN on both ends of an 802.1q trunk link. Failure to do so can cause bridging loops and VLAN leaks.
- All switches in a fabric cluster or logical chassis cluster must be configured with the same VLAN number.

Configuring and managing 802.1Q VLANs

Understanding the default VLAN configuration

The following table summarizes the default VLAN configuration. Consider this when making configuration changes.

TABLE 6 Default VLAN configuration

Parameter	Default setting
Default VLAN	VLAN 1
Interface VLAN assignment	All interfaces assigned to VLAN 1
VLAN state	Active
MTU size	2500 bytes

NOTE
Enter the **copy running-config startup-config** command to save your configuration changes.

Configuring interfaces to support VLANs

This section details the various tasks required to configure and manage VLAN traffic.

Enabling and disabling an interface port

NOTE
DCB interfaces are enabled by default in Brocade VCS Fabric mode.

NOTE
DCB interfaces do not support auto-negotiation of Ethernet link speeds. The DCB interfaces support 10-gigabit Ethernet and 1-gigabit Ethernet.

To enable and disable an interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

Configuring the MTU on an interface port

NOTE
The entire fabric acts like a single switch. Therefore, MTU is applicable only on the edge-ports, and not on ISL.

To configure the maximum transmission unit (MTU) on an interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **mtu** command to specify the MTU value on the interface port.

```
switch(conf-if-te-0/1)# mtu 4200
```

Creating a VLAN

On Brocade VDX hardware, VLANs are treated as interfaces from a configuration point of view.

By default all the DCB ports are assigned to VLAN 1 (VLAN ID equals 1). The *vlan_ID* value can be 1 through 4095. Refer to VLAN configuration guidelines and restrictions on page 73 for additional information.

To create a VLAN interface, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface vlan** command to assign the VLAN interface number.

```
switch(config)# interface vlan 1010
```

Enabling STP on a VLAN

Once all of the interface ports have been configured for a VLAN, you can enable Spanning Tree Protocol (STP) for all members of the VLAN with a single command. Whichever protocol is currently selected is used by the VLAN. Only one type of STP can be active at a time.

A physical interface port can be a member of multiple VLANs. For example, a physical port can be a member of VLAN 1015 and VLAN 55 simultaneously. In addition, VLAN 1015 can have STP enabled and VLAN 55 can have STP disabled simultaneously.

To enable STP for a VLAN, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **protocol spanning tree** command to select the type of STP for the VLAN.

```
switch(config)# protocol spanning tree mstp
```

3. Enter the **interface** command to select the VLAN interface number.

```
switch(config)# interface vlan 1015
```

4. Enter the **no spanning-tree shutdown** command to enable spanning tree on VLAN 1015.

```
switch(conf-if-vl-1015)# no spanning-tree shutdown
```

Disabling STP on a VLAN

Once all of the interface ports have been configured for a VLAN, you can disable STP for all members of the VLAN with a single command.

To disable STP for a VLAN, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to select the VLAN interface number.

```
switch(config)# interface vlan 55
```

3. Enter the **spanning-tree shutdown** command to disable STP on VLAN 55.

```
switch(conf-if-vl-55)# spanning-tree shutdown
```

Configuring an interface port as a Layer 2 switch port

To configure the interface as a Layer 2 switch port, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **switchport** command to configure the interface as a Layer 2 switch port.

```
switch(config-if-te-1/0/1)# switchport
```

4. Enter the **do show** command to confirm the status of the DCB interface.

```
switch(conf-if-te-1/0/1)# do show interface tengigabitethernet 1/0/1
```

5. Enter the **do show** command to confirm the status of the DCB interface running configuration.

```
switch(conf-if-te-1/0/1)# do show running-config interface tengigabitethernet 1/0/1
```

Configuring an interface port as an access interface

Each DCB interface port supports admission policies based on whether the frames are untagged or tagged. Access mode admits only untagged and priority-tagged frames.

To configure the interface as an access interface, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **switchport** command to make the interface a Layer 2 switch port.

```
switch(conf-if-te-0/1)# switchport
```

4. Configure the interface as an access interface.

```
switch(conf-if-te-0/1)# switchport mode access
```

5. Enter the **switchport** command again to configure the DCB interface as a VLAN.

```
switch(conf-if-te-0/1)# switchport access vlan 20
```

Configuring an interface port as a trunk interface

Each DCB interface port supports admission policies based on whether the frames are untagged or tagged. Trunk mode admits only VLAN-tagged frames.

To configure the interface as a trunk interface, run the following steps in privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the interface port.

```
switch(config)# interface tengigabitethernet 1/0/19
```

3. Enter the **switchport** command to place the DCB interface into trunk mode.

```
switch(conf-if-te-1/0/19)# switchport mode trunk
```

4. Specify whether all, one, or none of the VLAN interfaces are allowed to transmit and receive through the DCB interface. Enter the one of the following commands as is appropriate for your needs.

Allowing only one VLAN to transmit or receive through the DCB interface

This example allows VLAN 30 to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan add 30
```

Allowing all VLANs to transmit or receive through the DCB interface

This example allows all VLANs to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan all
```

Excluding a VLAN from the DCB interface

This example allows all except VLAN 11 to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan except 11
```

Blocking all VLANs from the DCB interface

This example allows none of the VLANs to transmit or receive through the DCB interface.

```
switch(conf-if-te-1/0/19)# switchport trunk allowed vlan none
```

Disabling a VLAN on a trunk interface

To disable a VLAN on a trunk interface, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

   ```
   switch(config)# interface tengigabitethernet 1/0/10
   ```

3. Enter the **switchport** command to place the DCB interface into trunk mode.

   ```
   switch(conf-if-te-1/0/10)# switchport mode trunk
   ```

4. Enter the **switchport** command again to remove the VLAN ranges from the trunk port.

   ```
   switch(conf-if-te-1/0/10)# switchport trunk allowed vlan remove 30
   ```

Configuring protocol-based VLAN classifier rules

You can configure VLAN classifier rules to define specific rules for classifying frames to selected VLANs based on protocol and MAC addresses. Sets of rules can be grouped into VLAN classifier groups (refer to Deleting a VLAN classifier rule on page 78).

VLAN classifier rules (1 through 256) are a set of configurable rules that reside in one of these categories:

- 802.1Q protocol-based classifier rules
- Source MAC address-based classifier rules
- Encapsulated Ethernet classifier rules

NOTE

Multiple VLAN classifier rules can be applied per interface, provided that the resulting VLAN IDs are unique for the different rules.

802.1Q protocol-based VLANs apply only to untagged frames, or frames with priority tagging.

With both Ethernet-II and 802.2 SNAP encapsulated frames, the following protocol types are supported:

- Ethernet hexadecimal (0x0000 through 0xffff)
- Address Resolution Protocol (ARP)
- Fibre Channel over Ethernet (FCoE)
- FCoE Initialization Protocol (FIP)
- IP version 4 (IPv4)
- IP version 6 (IPv6)

NOTE
For complete information on all available VLAN classifier rule options, refer to the *Network OS Command Reference*.

Configuring a VLAN classifier rule

To configure a ARP protocol-based VLAN classifier rule, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **vlan classifier rule** command to configure a protocol-based VLAN classifier rule.

```
switch(config)# vlan classifier rule 1 proto ARP encap ethv2
```

NOTE
Refer to the *Network OS Command Reference* for complete information on all the protocols available for the **vlan classifier rule** command.

Configuring MAC address-based VLAN classifier rules

To configure a MAC address-based VLAN classifier rule, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to change to global configuration mode.

```
switch# configure terminal
```

2. Enter the **vlan classifier rule** command to configure a MAC address-based VLAN classifier rule.

```
switch(config)# vlan classifier rule 5 mac 0008.744c.7fid
```

Deleting a VLAN classifier rule

VLAN classifier groups (1 through 16) can contain any number of VLAN classifier rules.

To configure a VLAN classifier group and remove a VLAN classifier rule, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Specify a VLAN classifier group and delete a rule.

```
switch(config)# vlan classifier group 1 delete rule 1
```

Creating a VLAN classifier group and adding rules

VLAN classifier groups (1 through 16) can contain any number of VLAN classifier rules.

To configure a VLAN classifier group and add a VLAN classifier rule, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Create a VLAN classifier group and add a rule.

```
switch(config)# vlan classifier group 1 add rule 1
```

Activating a VLAN classifier group with an interface port

To associate a VLAN classifier group with an interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.

2. Enter the **interface** command to specify the DCB interface type and RBridge/slot/port number.

```
switch(config)# interface tengigabitethernet 22/0/10
```

3. Enter the **vlan classifier** command to activate and associate it with a VLAN interface (group 1 and VLAN 2 are used in this example).

```
switch(conf-if-te-22/0/10)# vlan classifier activate group 1 vlan 2
```

NOTE
This example assumes that VLAN 2 was already created.

Displaying VLAN information

NOTE
The **show vlan brief** command displays the VLAN as inactive if there are no member ports associated to that VLAN, or if the ports associated are in an admin down state.

To display VLAN information, perform one or both the following steps from privileged EXEC mode.

1. To display the configuration and status of the specified interface, enter the **show interface** command.

```
switch# show interface tengigabitethernet 3/0/10
```

2. To display the specified VLAN information, enter the **show vlan** command.

   For example, this syntax displays the status of VLAN 20 for all interfaces, including static and dynamic:

```
switch# show vlan 20
```

Configuring the MAC address table and conversational MAC learning

Each DCB port has a MAC address table that stores the source MAC address of all frames. In addition, Brocade VDX hardware has a configurable aging timer. If a source MAC address remains inactive for a specified number of seconds, it is removed from the address table. For detailed information on how the switch handles MAC addresses in a Layer 2 Ethernet environment, refer to FCoE and Layer 2 Ethernet on page 40.

In addition, support is now provided for conversational MAC learning (CML).

Conversational MAC learning

Layer 2 switches use forwarding tables to direct traffic to specific ports, based on the VLAN number and destination MAC address of the frame. When there is no entry corresponding to the destination MAC address in the incoming VLAN, the frame is sent to all forwarding ports within the respective VLAN, which causes flooding. MAC address learning is an essential Layer 2 feature whereby the source MAC addresses of each received packet is stored so that future packets destined for that address can be forwarded only to the bridge interface on which the address is located.

Prior to Network OS v5.0.0, each node in the VCS Fabric stores in its hardware table the Layer 2 addresses of all end stations that are learned, without consideration for actual conversations. This poses a considerable unnecessary burden on system memory. Beginning with Network OS v5.0.0, the global **mac-address-table** command has been enhanced with a **conversational** keyword, allowing the user to enable conversational MAC (address) learning, or CML, globally on a switch.

Specifying or disabling the aging time for MAC addresses

You can set the length of time that a dynamic entry remains in the MAC address table after the entry is used or updated. Static address entries are never aged or removed from the table. You can also disable the aging time. The default is 300 seconds.

> NOTE
> To disable the aging time for MAC addresses, enter an aging time value of 0.

To specify an aging time or disable the aging time for MAC addresses, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the appropriate command based on whether you want to specify an aging time or disable the aging time for MAC addresses:

```
switch(config)# mac-address-table aging-time 600
```

Adding static addresses to the MAC address table

To add a static address to the MAC address table, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Add the static address 0011.2222.3333 to the MAC address table with a packet received on VLAN 100:

```
switch(config)# mac-address-table static 0011.2222.3333 forward tengigabitethernet 1/0/1 vlan 100
```

Enabling conversational MAC learning

You can disable dynamic MAC address learning on a switch, and enable CML, by means of the **mac-address-table learning-mode conversational** command.

> NOTE
> The ability to disable source MAC address learning on a per-port, per-VLAN basis constrains traffic flooding to only the ports that are part of a VLAN. Disabling traditional dynamic MAC learning prevents the MAC address table from being saturated. For example, when a device is being attacked by many packets with different source MAC address, the updating of the MAC address table is significantly impaired.

Do the following in global configuration mode to disable dynamic MAC learning and enable CML globally on an RBridge.

```
switch(config)# mac-address-table learning-mode conversational
```

Do the following to revert to legacy dynamic MAC learning mode.

```
switch(config)# no mac-address-table learning-mode
```

Do the following to configure the destination MAC address aging interval to 60 seconds.

```
switch(config)# mac-address-table aging-time conversational 60
```

Do the following to revert to the default aging interval of 300 seconds.

```
switch(config)# no mac-address-table aging-time conversational
```

Disabling source MAC learning on an interface

You can disable legacy dynamic MAC address learning on one or more VLANs on an interface. (CML is disabled by default.) VLANs range from 1 through 4090 for 802.1Q VLANs, and from 4096 through 8191 for VLANs in a Virtual Fabrics context.

To add a VLAN and range of VLANs to a switchport trunk:

```
switch(conf-if-te-4/0/5)# switchport trunk allowed vlan add 2000,3000-3500
```

To add the above VLANs to the MAC-learning-disabled list:

```
switch(conf-if-te-4/0/5)# mac-learning disable vlan add 2000,3000-3500
```

To remove a VLAN from the list:

```
switch(conf-if-te-4/0/5)# mac-learning disable vlan remove 2000
```

To disable MAC learning for all VLANs on the interface:

```
switch(conf-if-te-4/0/5)# no mac-learning disable
```

To disable MAC learning for VLANs 2000, 3000, and 3001 on the interface:

```
switch(conf-if-te-4/0/5)# mac-learning disable vlan 2000, 3000, 3001
```

To view the status of the VLANs, use the **show interface switchport** command, as in the following example:

```
switch# show interface tengigabitethernet 4/0/5 switchport

 Interface name                    : TenGigabitEthernet 4/0/5
 Switchport mode                   : trunk
 Fcoeport enabled                  : no
 Ingress filter                    : enable
 Acceptable frame types            : vlan-tagged only
 Native Vlan                       : 1
 Active Vlans                      : 1-2,4,7-10
 Inactive Vlans                    : -
 MAC learn disable Vlans           :  2000,3000,3001
```

Private VLANs

A private VLAN (PVLAN) domain is built with at least one pair of VLAN IDs; one (and only one) primary VLAN ID plus one or more secondary VLAN IDs. A primary VLAN is the unique and common VLAN identifier of the whole private VLAN domain and of all its VLAN ID pairs. Secondary VLANs can be configured as one of two types: either isolated VLANs or community VLANs. Up to 24 isolated or community VLANs can be part of a PVLAN domain.

An isolated VLAN is a secondary VLAN whose distinctive characteristic is that all hosts connected to its ports are isolated at Layer 2. A community VLAN is a secondary VLAN that is associated to a group of ports that connect to a designated community of end devices with mutual trust relationships.

A PVLAN is often used to isolate networks from security attacks, or to simplify IP address assignments.

Within the private VLAN, ports can be assigned port types. A port can be assigned to only one kind of port type at a time. The types of ports available for private VLANs are described in the following table.

TABLE 7 Private VLAN terms and definitions

Term	Description
Isolated port	An isolated port cannot talk to any other port in the private VLAN domain except for promiscuous ports and traffic ports. If a customer device needs to have access only to a gateway router, then it should be attached to an isolated port.

TABLE 7 Private VLAN terms and definitions (continued)

Term	Description
Community port	A community port is part of a group of ports that have Layer 2 communications with one another, and can also talk to any promiscuous port. For example, if you have two devices that you want to be isolated from other devices, but still be able to communicate between themselves, then community ports should be used. You cannot configure multiple community VLANs on a single port.
Promiscuous port	A promiscuous port can talk to all other types of ports. A promiscuous port can talk to isolated ports as well as community ports. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with the customer endpoints are typically connected using promiscuous ports.
Trunk port	A trunk port connects two switches and carries two or more VLANs.
Promiscuous trunk port	A promiscuous trunk port carries multiple primary and normal VLANs. Packets are received and transmitted with primary or regular VLAN tags. Otherwise, the port operates as a promiscuous port.
Secondary VLAN	A VLAN used to implement PVLANs. Secondary VLANs are associated with a primary VLAN, and carry traffic from hosts to other allowed hosts or routers.
Community VLAN	A secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways, and to other host ports in the same community. Multiple community VLANs are permitted in a PVLAN.
Primary VLAN	A PVLAN has only one primary VLAN. Every port in a PVLAN is a member of the primary VLAN. The primary VLAN carries unidirectional traffic downstream from the promiscuous ports to the isolated and community ports and to other promiscuous ports.

PVLAN configuration guidelines and restrictions

Follow these guidelines and restrictions when configuring VLANs:

- VE configuration is not supported on a primary VLAN.
- IGMP is not supported on private VLANs; however you can create an IGMP configuration. The configuration succeeds but the hardware is not programmed.
- For private VLANs, egress ACLs on the primary VLAN are applied only for the traffic that ingresses and egresses from the primary VLAN, and not for the traffic that gets translated from the secondary VLAN to the primary VLAN.
- For private VLANs, egress ACLs on the primary VLAN are also applied to the traffic that gets translated to the secondary VLAN.
- STP is not supported on private VLAN host ports.

Associating the primary and secondary VLANs

This procedure configures the PVLAN and associates the secondary VLAN with the primary VLAN.

1. Configure the VLAN interface.

```
switch(config)# interface vlan 10
```

2. Configure the VLAN as a primary PVLAN.

```
switch(conf-if-vl-10)# private-vlan primary
```

3. Create multiple community VLANs, for use in Step 5.

```
switch(config)# interface vlan 100
switch(conf-if-vl-100)# private-vlan community
```

4. Configure the secondary VLAN (isolated).

```
switch(config)# interface vlan 200
switch(conf-if-vl-200)# private-vlan isolated
```

5. Associate the multiple community VLANs and the isolated VLAN.

```
switch(config)# interface vlan 10
switch(conf-if-vl-10)# private-vlan association add 100,200
```

6. Exit VLAN configuration mode.

```
switch(conf-if-vl-10)# exit
```

Configuring an interface as a PVLAN promiscuous port

This procedure configures an interface as the PVLAN promiscuous port.

1. Specify the interface.

```
switch(config)# interface tengigabitethernet 0/1
```

2. Mark the interface as switch port

```
switch(conf-if-te-0/1)# switchport
```

3. Configure the interface as a PVLAN promiscuous port (untagged).

```
switch(conf-if-te-0/1)# switchport mode private-vlan promiscuous
```

4. Configure the interface as a PVLAN promiscuous port (tagged).

```
switch(conf-if-te-0/1)# switchport mode private-vlan trunk promiscuous
```

5. Associate the interface with a PVLAN.

```
switch(conf-if-te-0/1)# switchport private-vlan mapping 10 add 100,200
```

6. Configure a normal VLAN on the PVLAN promiscuous port.

```
switch(conf-if-te-0/1)# switchport trunk allowed vlan add 500
```

Configuring an interface as a PVLAN host port

This procedure configures an interface as the PVLAN host port and thereby isolates a VLAN.

1. Specify the interface.

```
switch(config)#interface tengigabitethernet 1/0/1
```

2. Mark the interface as a switch port.

```
switch(conf-if-te-1/0/1)# switchport
```

3. Configure the interface as a PVLAN host port that is tagged.

```
switch(conf-if-te-1/0/1)# switchport mode private-vlan trunk host
```

4. Alternatively, configure the interface as a PVLAN host port that is untagged.

```
switch(conf-if-te-1/0/1)# switchport mode private-vlan host
```

5. Associate the interface with a PVLAN and isolate VLAN 100.

```
switch(conf-if-te-1/0/1)# switchport private-vlan host-association 10 100
```

Configuring an interface as a PVLAN trunk port

This procedure configures an interface as a PVLAN trunk port.

1. Specify the interface.

```
switch(config)# interface tengigabitethernet 0/1
```

2. Mark the interface as switch port.

```
switch(conf-if-te-0/1)# switchport
```

3. Configure the interface as a PVLAN trunk port.

   Do not complete this step if the host is a plain, untagged server.

```
switch(conf-if-te-0/1)# switchport mode private-vlan trunk
```

4. Configure the association between primary VLANs and secondary VLANs and the PVLAN trunk port with a PVLAN.

```
switch(conf-if-te-0/1)# switchport private-vlan association trunk 10 100
```

   NOTE

   Multiple PVLAN pairs can be specified by means of the **switchport private-vlan association trunk** command, so that a PVLAN trunk port can carry multiple secondary VLANs. If an association is specified for the existing primary VLAN, the existing association is replaced. If there is no trunk association, any packets received on secondary VLANs are dropped.

5. Configure a normal VLAN on the PVLAN trunk port.

```
switch(conf-if-te-0/1)# switchport private-vlan trunk allowed vlan 400
```

6. Configure a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port. If there is no native VLAN configured, all untagged packets are dropped. If the native VLAN is a secondary VLAN and the port does not have the association for the secondary VLAN, the untagged packets are dropped.

```
switch(conf-if-te-0/1)# switchport private-vlan trunk native vlan 600
```

Displaying PVLAN information

To display private VLAN information, use the **show vlan private-vlan** command to see the private VLAN types (for example, "private," "isolated," and "community," as in the following example.

```
device# show vlan private-vlan
```

VXLAN Overlay Gateways for NSX Controller Deployments

Introduction to VXLAN overlay gateways with NSX Controller

Virtual Extensible LAN (VXLAN) is an overlay network that extends Layer 2 domains over Layer 3 networks. The overlay network supports elastic compute architectures, enabling network engineers to scale a cloud computing environment while logically isolating cloud applications and tenants.

VXLAN extends the virtual LAN (VLAN) address space by adding a 24-bit segment ID and increasing the number of available VLAN IDs to 16 million, in a Virtual Fabrics context. The VXLAN segment ID in each frame differentiates individual logical networks, allowing millions of isolated Layer 2 VXLAN networks to coexist on a common Layer 3 infrastructure. As with VLANs, only virtual machines (VMs) within the same logical network can communicate with each other.

VXLAN creates large-scale, isolated virtual Layer 2 networks for virtualized and multi-tenant environments by encapsulating frames in VXLAN packets. Frame encapsulation is performed by means of a VXLAN Network Identifier (VNI) tunnel endpoint (VTEP), which originates or terminates VXLAN tunnels.

Because not all devices and servers are capable of sending or receiving VXLAN traffic, a VXLAN overlay gateway allows communication between the VXLAN-aware world and the non-VXLAN-aware world. In the non-VXLAN-aware world, a broadcast domain represented by a VLAN typically comprises the virtual cluster switch and other switches and devices behind the switch.

In the initial phase, the VXLAN-aware world consists of virtual networks that are managed by a third-party system known as the VMware NSX Controller. The NSX Controller is a highly available distributed system that manages, or orchestrates, all network components and connections in a virtual network. The VXLAN overlay gateway must communicate with the NSX Controller to create tunnels with VXLAN-aware end devices. The NSX Controller function can comprise a cluster of controllers. The orchestrator function resides most commonly at top of rack (ToR); however, it can also be deployed as an aggregator.

Beginning with Network OSv5.0.0, MAC, IPv4, and IPv6 ingress ACLs are supported, as well as sFlow configurations.

NOTE

Note the following conditions for this feature:

- VXLAN overlay gateways are supported only on the Brocade VDX 6740 family and VDX 6940 family.
- VXLAN gateways must be in logical chassis cluster mode. This allows the virtual cluster switch to present itself as a single device to the NSX Controller, in conjunction with a VMware Hypervisor.

Beginning with Network OSv7.0.0, support is provided for the following:

- Redistribution (load balancing) of broadcast, unknown unicast, and multicast (BUM) VLANs across VXLAN NSX Service Node (SN) tunnels in a VCS Fabric. (Service Nodes are now referred to as *replicators*.)
- VLAN classification profiles in hardware. See "Configuring VLAN classification profiles" in the *Network OS Administration Guide*.

Beginning with Network OSv7.0.1, a default VLAN classification profile is provided to support NSX Controller deployments.

VXLAN NSX replicator load balancing

This feature enables the distribution of egress broadcast, unicast, and unknown multicast (BUM) traffic across tunnels to VMware NSX replicators, previously referred to as *Service Nodes (SNs)*.

When there are multiple replicator tunnels, the switch can distribute the BUM traffic across all such tunnels. This is done by assigning different set of VLANs to each of the tunnels. Egress BUM traffic on different VLANs are now forwarded through different tunnels, hence increasing the overall efficiency of the network. The switch can receive ingress BUM traffic over any VLAN on any replicator tunnel. When a tunnel is deleted or a Bidirectional Forwarding Detection (BFD) " down" state is detected, the system automatically reassigns its BUM VLANs to other available replicator tunnels.

Note the following considerations and limitations:

- This feature is supported on the Brocade VDX 6740 and VDX 6940 series. It is not supported on the Brocade VDX 8770 series.
- This feature is applicable only to NSX VXLAN tunnels, not to VXLAN overlay gateway L2-Extension VXLAN tunnels.
- BFD must be enabled on the NSX replicator.
- Momentary traffic disruptions can occur when new replicator tunnels come online. A system reboot is not required.
- Duplicate traffic can occur momentarily as replicator tunnels come online or go offline.
- VLANs can become "skewed" with respect to load balancing when the user removes specific ranges of VLANs from a given tunnel.
- You can manually trigger the redistribution of BUM VLANs on all NSX replicator tunnels by using the **tunnel replicator bum-vlans redistribute** command in privileged EXEC mode. Refer to the *Network OS Command Reference* for details on this command.
- This configuration cannot be changed if there are any replicator tunnels in the system. Because there is no option to remove only replicator tunnels, all tunnels must be removed. as follows: (1) remove all RBridges attached to the overlay gateway, (2) delete the overlay gateway configuration, and (3) delete the NSX Controller configuration.
- Replicator load balancing is enabled following a firmware upgrade from Network OS 6.x or 7.0.0 to Network OS 7.0.1. A "disabled" load-balancing configuration is lost during a downgrade from Network OS 7.0.1, and the feature is lost during a downgrade to Network OS 6.x.

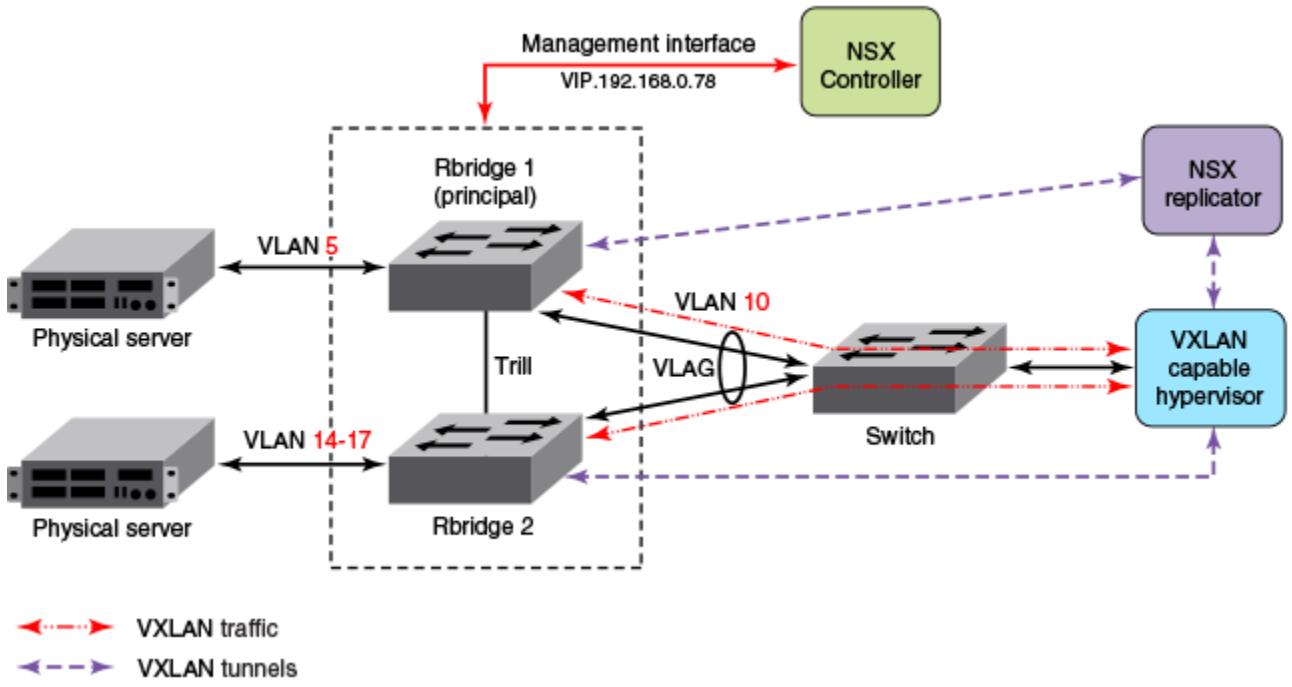
Configuring a VXLAN overlay gateway for NSX Controller deployments

Before you configure the VXLAN overlay gateway, complete the configuration steps in Configuring VRRP-E for NSX Controller deployments on page 88. The prerequisite steps demonstrate how to configure the RBridges as part of a Virtual Router Redundancy Protocol - Extended (VRRP-E) group, which is a requirement for a VXLAN overlay gateway. Also, you must configure the identical virtual Ethernet (VE) and VRRP-E group on all the RBridges for the VXLAN overlay gateway.

High-level communication in a VXLAN environment with an NSX Controller

The following illustration provides a basic view of the interaction of components in a VXLAN environment that uses an NSX Controller.

FIGURE 11 High-level communication for VXLAN overlay gateway with NSX Controller



A maximum of four RBridges are supported in a VXLAN-enabled VCS Cluster. The VXLAN Gateway option should be enabled on all the RBridges of the cluster.

In the example topology shown above, RBridge 1 and RBridge 2 make up a two-node cluster with the VXLAN gateway function enabled on both RBridges. The RBridges in the VCS cluster are connected to a VXLAN-capable hypervisor through a Layer 2 device. The VXLAN tunnel traffic is transported over VLAN 10 between the VCS and the VXLAN-capable hypervisor. An NSX replicator communicates with RBridges through VXLAN tunnels. Additionally, there are two physical servers connected to the VCS cluster. The VXLAN gateway-enabled RBridges transmit the VXLAN traffic from the hypervisor, as well as the VLAN-based traffic from the physical servers.

For the control communication, the principal switch of the VXLAN overlay gateway communicates with the NSX Controller. This communication occurs over the management interface (depicted by the red line in the illustration above). In this example, RBridge 1 is the principal RBridge for the VCS cluster.

Coordination of activities in NSX Controller deployments

Be sure to coordinate your activities with the administrators of the virtual network and NSX Controller to help ensure a successful setup. This includes providing the NSX administrator with an inventory of switches, ports, and VLANs in your cluster.

The NSX administrator creates the virtual network, assigns a VXLAN Network Identifier (VNI) to this network, and selects the ports that are to be attached to this virtual network. If ports on the virtual cluster switch are selected, the NSX Controller pushes network information to the switch. This information includes VNI, VLAN-to-VNI bindings for each port, and MAC-VNI-VTEP mappings for each of the MAC addresses in the virtual network. The NSX Controller is not aware of MAC addresses in the Layer 2 network behind the switch at this time.

Configuring VRRP-E for NSX Controller deployments

The steps that follow show an example VRRP-Extended (VRRP-E) group configuration for the RBridges shown in Figure 11 on page 87. VRRP-E is required to be configured on all the RBridges of the VCS based VXLAN gateway for NSX.

> NOTE
> VRRP-E is necessary as the VXLAN tunnel termination and redundancy are related to the VRRP-E vMAC on the Brocade VDX 6740.

In the example shown in shown in Figure 11 on page 87, the VRRP-E functionality is configured on the VE interface for transport VLAN 10, which carries the VXLAN traffic between the VCS cluster and the VXLAN-capable hypervisor on each of the RBridges in the VCS cluster.

> NOTE
> The existence of a VTEP for VXLAN bridging does not affect any configured routing and switching performed by the RBridges in the VCS cluster for non-VXLAN traffic.

1. Enter global configuration mode:

```
switch# config
```

2. Enter RBridge ID configuration mode for the first RBridge in the logical chassis cluster (this example uses RBridge ID 1, as in the example topology).

```
switch(config)# rbridge-id 1
```

3. Enable VRRP-E for this RBridge.

```
switch(config-rbridge-id-1)# protocol vrrp-extended
```

4. Enter the **interface ve** command to configure a virtual Ethernet (VE) interface for RBridge 1 (for example, 10) that corresponds to an already created VLAN, and enter the IP address and mask for the interface (for example, 10.60.60.3/24).

```
switch(config-rbridge-id-2)# interface ve 10
```

5. Enter the IP address and mask for the interface (for example, 10.10.10.3/24).

```
switch(config-Ve-10)# ip address 10.10.10.3/24
```

6. Enter the **no shutdown** command to enable the interface.

```
switch(config-Ve-10)# no shutdown
```

7. Enter the **vrrp-extended-group** command for the group ID (100 in this example) of the VRRP-E group.

```
switch(config-Ve-10)# vrrp-extended-group 100
```

8. Assign a virtual MAC address by entering the **virtual-mac** command.

```
switch(config-vrrp-extended-group-100)# virtual-mac 02e0.5200.00xx
```

9. Enter a virtual IP address of the VRRP-E group, as in the following example.

```
switch(config-vrrp-extended-group-100)# virtual-ip 10.10.10.230
```

10. Enable short-path forwarding on the virtual router.

```
switch(config-vrrp-extended-group-100)# short-path-forwarding
```

11. Exit this configuration mode and enter global configuration mode.

```
switch(config-vrrp-extended-group-100)# end
switch# configure
```

12. Enter RBridge ID configuration mode for the other RBridge in your logical chassis cluster (RBridge 2 in the example topology).

```
switch(config)# rbridge-id 2
```

13. Enable VRRP-E for this RBridge.

```
switch(config-rbridge-id-2)# protocol vrrp-extended
```

14. Enter the **interface ve** command to configure the same VE interface as for RBridge 1.

```
switch(config-rbridge-id-2)# interface ve 10
```

15. Enter the IP address and mask for RBridge 2 for this VE interface (for example, 10.60.60.4/24).

```
switch(config-Ve-10)# ip address 10.10.10.4/24
```

16. Enter the **no shutdown** command to enable the interface.

```
switch(config-Ve-10)# no shutdown
```

17. Enter the **vrrp-extended-group** command with the group ID of the VRRP-E group used on the other RBridge.

```
switch(config-Ve-10)# vrrp-extended-group 100
```

18. Enter the **virtual-mac** command and assign the virtual MAC address used on RBridge 1.

```
switch(config-vrrp-extended-group-100)# virtual-mac 02e0.5200.00xx
```

19. Enter a virtual IP address for the VRRP-E group, as in the following example.

```
switch(config-vrrp-extended-group-100)# virtual-ip 10.10.10.230
```

20. Enable short-path forwarding on the virtual router.

```
switch(config-vrrp-extended-group-100)# short-path-forwarding
```

Configuring a loopback interface VTEP for NSX Controller deployments

As an alternative to the VRRP-E method, you can configure a loopback interface to serve as a VTEP.

Do the following to configure a loopback interface as a VTEP.

NOTE
You must manually configure distinct router IDs, by means of the **ip router-id** command, for use by routing protocols.

1. Enter VXLAN overlay gateway configuration mode.

```
switch(config)# overlay-gateway gateway1
```

2. Use the **ip interface** command to specify a loopback interface.

```
switch(config-overlay-gw-gateway1)# ip interface loopback 25
```

NOTE
When a VXLAN gateway is active (as configured by means of the **activate** command in VXLAN overlay gateway configuration mode), the loopback interface cannot be deleted. You must first use the **no activate** command.

VXLAN gateway and NSX Controller deployments

Consider the following guidelines before configuring the VXLAN gateway.

The following rules apply to VXLAN traffic packets transmitted by VXLAN to VLAN bridging:

- If the VXLAN packet entering a VDX VTEP-enabled device on the VLAN for which the VRRP-E session is addressed to the VRRP-E virtual mac, and the final destination is an NSX-configured VXLAN tunnel (as identified by the tunnel parameters of source IP and destination IP), then the VXLAN traffic is a candidate for VXLAN to VLAN bridging.
- If the VXLAN packet entering a VDX VTEP-enabled device on a Layer 3 interface (such as a routing next hop) that is different from the VRRP-E based VE interface configured for the VTEP, and its final destination is an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of source IP and destination IP), then the VXLAN traffic is routed to the VTEP interface in the VDX and is a candidate for VXLAN to VLAN bridging.
- If the VXLAN packet entering a VDX VTEP-enabled device on a Layer 3 interface (such as a routing next hop) is different from the VRRP-E based-VE interface configured for the VTEP, but at an ingress interface where the VTEP VRRP-E VLAN is also configured, and the final destination is an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of source IP and destination IP), then the VXLAN traffic is routed to the VTEP interface in the VDX and is a candidate for VXLAN to VLAN bridging, but only if the destination mac of the ingressing VXLAN traffic is the same as that of the virtual mac of the VTEP VRRP-E session. This occurs by creating VE interfaces for each of the ingressing transport VLANs on all the RBridges in the VCS, and then configuring them with the same VRRP-E VRID and virtual-mac address as the VRRP-E VRID and virtual-mac address that was configured for the VTEP.

Configuring the VXLAN gateway for NSX Controller deployments

The following steps detail how to configure a VXLAN overlay gateway and point it to the NSX Controller. This procedure references Figure 11 on page 87. Both the NSX for Multi-Hypervisor (NSX-MH) and the NSX for vSphere (NSX-V) are supported. The configuration for an NS-MH requires TCP port 6632 to support the Open vSwitch Database Management Protocol (OVSDB) management channel. The configuration for an NSX-V requires TCP port 6640 to support the OVSDB management channel.

Perform all the following steps on the principal switch (RBridge 1 in the example topology).

NOTE
A tunnel will not be created if there is not an active VM on the Hypervisor. If the tunnel is not created, check the VM connectivity on the Hypervisor.

1. The following substeps create a VXLAN Network Identifier (VNI) tunnel endpoint (VTEP).
   a) Enter global configuration mode, then enter the **overlay-gateway** *name* **type hardware-vtep** command.

      NOTE
      The **type hardware-vtep** keywords are required to specify that this deployment uses an NSX Controller (this keyword also supports OpenStack deployments. The name "gateway1" is only an example; this can be a name of your choice.

```
device# config
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)# type hardware-vtep
```

   b) Enter the **attach rbridge-id** command to attach existing RBridge IDs to this VXLAN gateway instance. You can specify a range of RBridge IDs up to a maximum of four.

```
device(config-overlay-gw-gateway1)# attach rbridge-id add 1-2
```

c) Enter the **ip interface ve** *veid* **vrrp-extended-group** *group-ID* command to set the IP address of the VXLAN overlay gateway, as shown in the following example.

```
device(config-overlay-gw-gateway1)# ip interface ve 10 vrrp-extended-group
100
```

The command accepts the VE interface ID and VRPP-E group ID, then sets the VXLAN overlay gateway's IP address as identical to the already configured VRRP-E virtual IP address. Tunnels that form use this IP address as the source IP address for outgoing packets.

d) Enter the **attach vlan** *vlan_id* command to export the desired VLANs (these are VLANs that can be mapped to VXLAN domains), as shown in the following example.

```
device(config-overlay-gw-gateway1)# attach vlan 5,14-17
```

NOTE

Virtual Fabrics cannot be attached when the overlay gateway type is **hardware-vtep** and Virtual Fabrics cannot be extended.

All the MAC addresses that the VXLAN overlay gateway learns on these VLANs are shared with the NSX Controller. When a MAC address ages out in VCS, the MAC address is removed from the NSX Controller.

NOTE

There is also an option to list specific MAC addresses. In this case, other MAC addresses that are learned for the VLAN are not shared with the NSX Controller. For more information, refer to the **attach vlan** command in the *Network OS Command Reference*.

e) Optional: (Optional) You can enter the **enable statistics direction** command to enable statistics collection for tunnels you specify, as shown in the following example.

```
device(config-overlay-gw-gateway1)# enable statistics direction both vlan
add 14-17
```

This example command enables statistics collection for tunnels in both directions (transmitting and receiving) for the specified VLANs.

f) Optional: If you have created a SPAN destination outside of the VXLAN overlay gateway (as a monitor session), you can enter the **monitor session** command to monitor session traffic, as shown in the following examples.

```
device(config-overlay-gw-gateway1)# monitor session 1 direction both
remote-endpoint 1.2.3.4 vlan add 41-43

device(config-overlay-gw-gateway1)# monitor session 1 direction both
remote-endpoint any vlan add 41-43
```

g) Enter the **activate** command to activate this gateway instance.

```
device(config-overlay-gw-gateway1)# activate
```

This enables all tunnels associated with this gateway. VXLAN tunnels are not user configurable.

h) Return to privileged EXEC mode.

```
device(config-overlay-gw-gateway1)# end
device(config)# end
device#
```

2. Generate the security certificate for the VXLAN overlay gateway by entering the **nsx-controller client-cert generate** command.

NOTE
Certificate generation is a one-time-only action.

```
device# nsx-controller client-cert generate
```

3. In privileged EXEC mode, display the certificate by entering the **show nsx-controller client-cert** command, then provide the certificate to the NSX administrator.
4. The following substeps configure the management interface (depicted by the red line in Figure 11 on page 87), which allows communication between the VXLAN gateway and the NSX Controller:
   a) Enter global configuration mode.

```
device# config
```

   b) Enter the **vcs virtual ip address** command and assign an IP address and mask.

```
device(config)# vcs virtual ip address
192.168.0.78/24
```

   c) Enter the **nsx-controller** *name* command to specify a name for a new NSX Controller connection profile:

```
device(config)# nsx-controller profile1
```

   d) Enter the **ip address** command to set the IP address of the controller, the port, and connection-method settings for an NSX Controller connection profile as shown in this example.

NOTE
This example illustrates a configuration for the NSX-MH, which requires port 6632 to support the OVSDB management channel. Use port 6640 when configuring support for an NSX-V.

```
device(config-nsx-controller-profile1)# ip address 10.21.83.188 port 6632
```

   e) Optional: You can change the reconnect interval between the NSX Controller and the VCS Fabric in case the connection is lost. The default is 10 seconds, meaning that a reconnection is attempted every 10 seconds. To change this interval to 40 seconds, for example, use the **reconnect-interval** command:

```
device(config-nsx-controller-profile1)# reconnect-interval 40
```

   f) Finally, enter the **activate** command to activate the NSX Controller profile.

```
device(config-nsx-controller-profile1)# activate
```

   This command initiates the connection between the NSX Controller and the VCS Fabric.

NOTE

The following rules apply to a VXLAN packet entering an interface on a VTEP-enabled Brocade VDX switch for that packet to be a candidate for VXLAN-to-VLAN bridging:

- If the VXLAN packet is entering a VTEP-enabled switch on the VLAN for which the VRRP-E session is configured, and the packet is destined to the VRRP-E virtual MAC address and belongs to an NSX-configured VXLAN tunnel (as identified by the source and destination IP address in the VXLAN packet), then the VXLAN packet is a candidate for VXLAN-to-VLAN bridging.
- If the VXLAN packet is entering a VTEP-enabled switch at a Layer 3 interface (as a routing next hop) that is different from the VRRP-E based VE interface configured for the VTEP, and the packet belongs to an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of the source and destination IP addresses), then the VXLAN traffic is routed to the VTEP interface in the Brocade VDX and is a candidate for VXLAN-to-VLAN bridging, with the exception noted below.
- If the VXLAN packet entering a VTEP-enabled switch at a Layer 3 interface (as a routing next hop) that is different from the VRRP-E-based VE interface configured for the VTEP but is an ingress interface where the VTEP VRRP-E VLAN is also configured, and the packet belongs to an NSX-configured VXLAN tunnel (as identified by the VXLAN tunnel parameters of the source and destination IP addresses), then the VXLAN traffic is routed to the VTEP interface in the Brocade VDX and is a candidate for VXLAN-to-VLAN bridging, only if the destination MAC address of the ingressing VXLAN packet is the same as that of the virtual MAC address of the VTEP VRRP-E session. This use case is addressed by creating VE interfaces for each of the ingressing transport VLANs on all the RBridges in the VCS Fabric and configuring them with the same VRRP-E VRID and virtual MAC address as the VRRP-E VRID and virtual MAC address configured for the VTEP.

Configuring VXLAN NSX replicator load balancing

You can manually trigger the redistribution of broadcast/unicast/multicast (BUM) VLANs on all NSX replicator tunnels. This feature is supported on the Brocade VDX 6740 and VDX 6940 series. It is not supported on the Brocade VDX 8770 series.

For details, see the section "VXLAN NSX replicator load balancing" in this chapter.

1. In privileged EXEC mode, enter the **tunnel replicator bum-vlans redistribute** command.

```
device# tunnel replicator bum-vlans redistribute
```

2. To view details of BUM-enabled replicator tunnels, as well as the BUM VLANs (in range format) for each tunnel, use the **show tunnel replicator** command as in the following example:

```
device# show tunnel replicator
Tunnel 61442, mode VXLAN, rbridge-ids 1
Ifindex 2080436226, Admin state up, Oper state up, BFD up
Overlay gateway "GW1", ID 1
Source IP 20.20.1.1 ( Loopback  11), Vrf default-vrf
Destination IP 20.20.0.197
Configuration source VTEP Controller
MAC learning disabled
BUM vlans 41,1414-1813 (401 vlans)
Active next hops on rbridge 1:
    IP: 19.1.0.2, Vrf: default-vrf
    Egress L3 port: Te 1/0/15, Outer SMAC: 0027.f8db.e068
    Outer DMAC: 0027.f83a.349d
    Egress L2 Port: Te 1/0/15, Outer ctag: 0, stag:0, Egress mode: Local
    BUM forwarder: no

Packet count: RX 11441           TX 0
Byte count  : RX (NA)            TX 0
```

```
Tunnel 61443, mode VXLAN, rbridge-ids 1
Ifindex 2080436227, Admin state up, Oper state up, BFD up
Overlay gateway "GW1", ID 1
Source IP 20.20.1.1 ( Loopback  11), Vrf default-vrf
Destination IP 20.20.0.181
Configuration source VTEP Controller
MAC learning disabled
BUM vlans 42,1814-2213 (401 vlans)
Active next hops on rbridge 1:
    IP: 19.1.0.2, Vrf: default-vrf
    Egress L3 port: Te 1/0/15, Outer SMAC: 0027.f8db.e068
    Outer DMAC: 0027.f83a.349d
    Egress L2 Port: Te 1/0/15, Outer ctag: 0, stag:0, Egress mode: Local
    BUM forwarder: no

Packet count: RX 11436           TX 0
Byte count  : RX (NA)            TX 0
```

Additional commands for VXLAN configuration

Additional commands that support VXLAN configuration are listed in the following table.

NOTE

For complete information on the those commands as well as other VXLAN overlay-gateway commands and commands related to the NSX Controller, refer to the *Network OS Command Reference*.

TABLE 8 Additional commands for VXLAN configuration

Command	Description
clear overlay-gateway	Clears counters for the specified gateway.
enable statistics	Enables statistics for tunnels.
sflow remote-endpoint	Applies an sFlow profile for a VXLAN overlay gateway and sets the remote endpoints for tunnel interfaces.
show nsx controller	Displays connection status of the NSX Controller. Includes an option to display the gateway certificate that is needed for NSX "transport node" configuration.
show overlay-gateway	Displays status and statistics for the VXLAN overlay-gateway instance.
show running-config overlay-gateway	Displays the running configuration of the overlay gateway configuration, including the connection type.
show tunnel	Displays tunnel statistics, including those for the NSX Service Node.

Distributed VXLAN Gateways

Distributed VXLAN gateways overview

The distributed VXLAN gateways feature eliminates the need for an external gateway device.

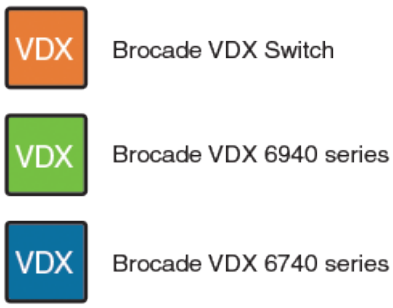
Prior to Network OS 6.0.1, VXLAN gateways had to be connected to the VCS Fabric (for example, a four-node gateway fabric) as an external device in order to bridge the overlay network and the physical networks that are interconnected by the VCS Fabric. Beginning with the current release, the gateway function can be hosted by the VCS RBridges. This eliminates the need for an external gateway device and optimizes network resources, improving network performance in the data center.

NOTE

Existing VRRP-E implementations cannot support more than four RBridges in a session. As a result, VRRP-E-based VXLAN gateways are also limited to a maximum of four RBridges.

The following sections present various use cases, both supported and unsupported, and their corresponding topologies. Refer to the following legend for those topologies.

FIGURE 12 Distributed VXLAN gateways legend



NOTE

In the initial release, this feature supported only Virtual Fabrics Extension deployments. Beginning with Network OS 7.0.1, support is provided for NSX Controller deployments.

Distributed VXLAN gateways supported topologies

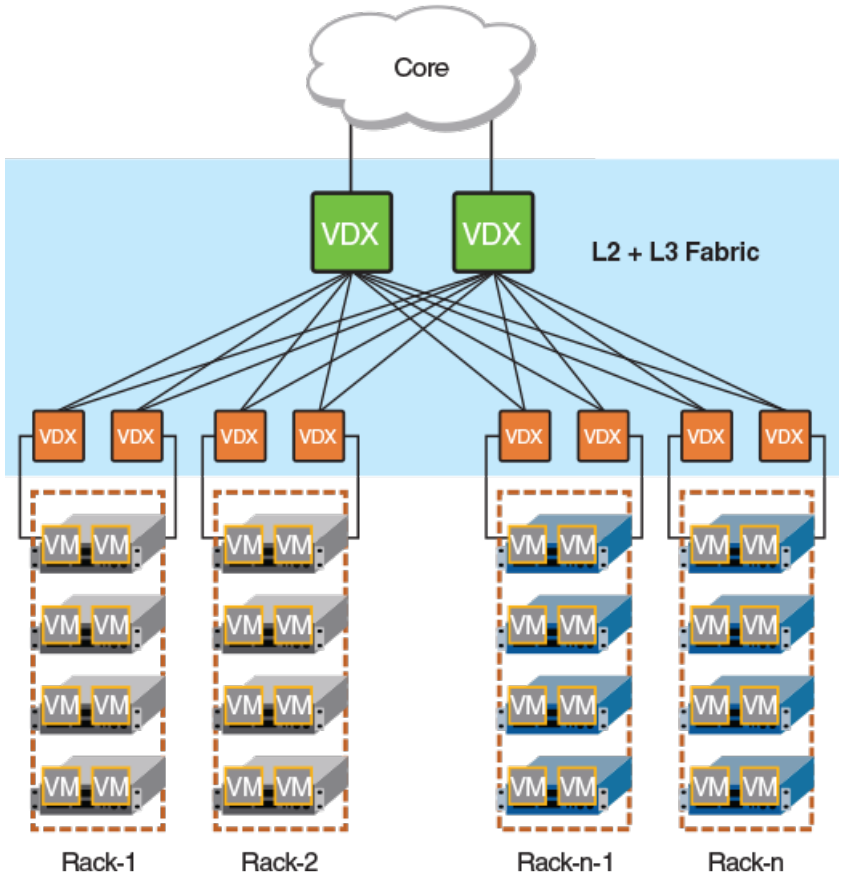
The following sections present the topologies that are supported in this release.

VDX 6940 at the aggregation layer

A Brocade VDX 6940 at the aggregation layer provides gateway functions for a VXLAN hypervisor at top of rack (ToR) or in the core.

A distributed VXLAN gateway makes it possible to connect to a hypervisor through a TRILL fabric, as the Brocade VDX 6940 supports TRILL-plus-VXLAN encapsulation. Refer to the following figure.

FIGURE 13 VDX 6940 in a Layer 2/Layer 3 fabric



In this topology, bridging a physical server and VXLAN servers that are located in the same rack requires interaction with an aggregation gateway, introducing an extra hop. The gateway supports VXLAN Network Identifier (VNI) classification for both east-west and north-south traffic. Note the following considerations:

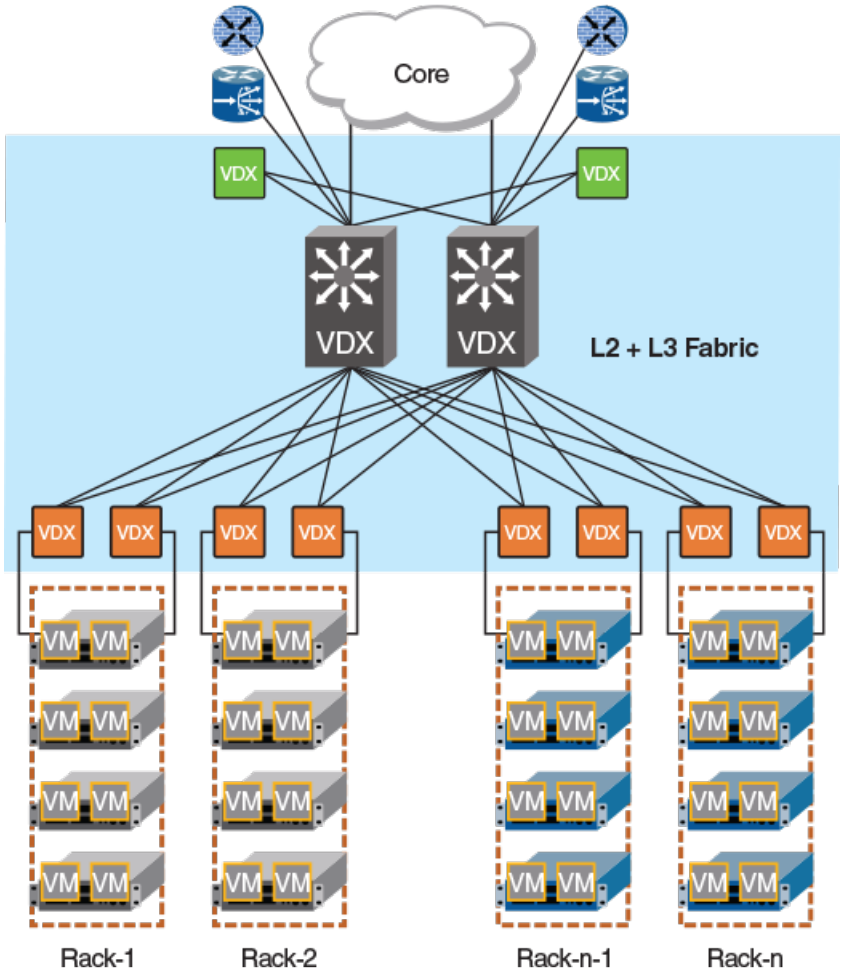
- The number of overlay networks that are supported in the fabric is limited by ASIC resources for the VNI classifications.
- The Brocade VDX 6940 is not supported at top of rack. This prevents VLAN-to-VXLAN traffic from "tromboning" in case the ToR gateway that does the VXLAN encapsulation is not in the same rack that holds the VXLAN server.

VDX 6940 as an appliance

A Brocade 6940 gateway can be inserted into the fabric as an appliance.

The following figure illustrates how a data center can use a VXLAN-incapable aggregation switch (such as a Brocade VDX 8770) to provide connectivity to the core while the switch is attached to a VXLAN gateway functioning as an appliance. Traffic between the physical server and a VXLAN-enabled server must always take an extra hop through the VDX to reach the appliance gateway.

FIGURE 14 VDX 6940 as a gateway appliance



Distributed VXLAN gateways unsupported topologies

The following topologies, although they can provide a certain degree of functionality, are not supported by Brocade for this feature, for reasons detailed in the sections below.

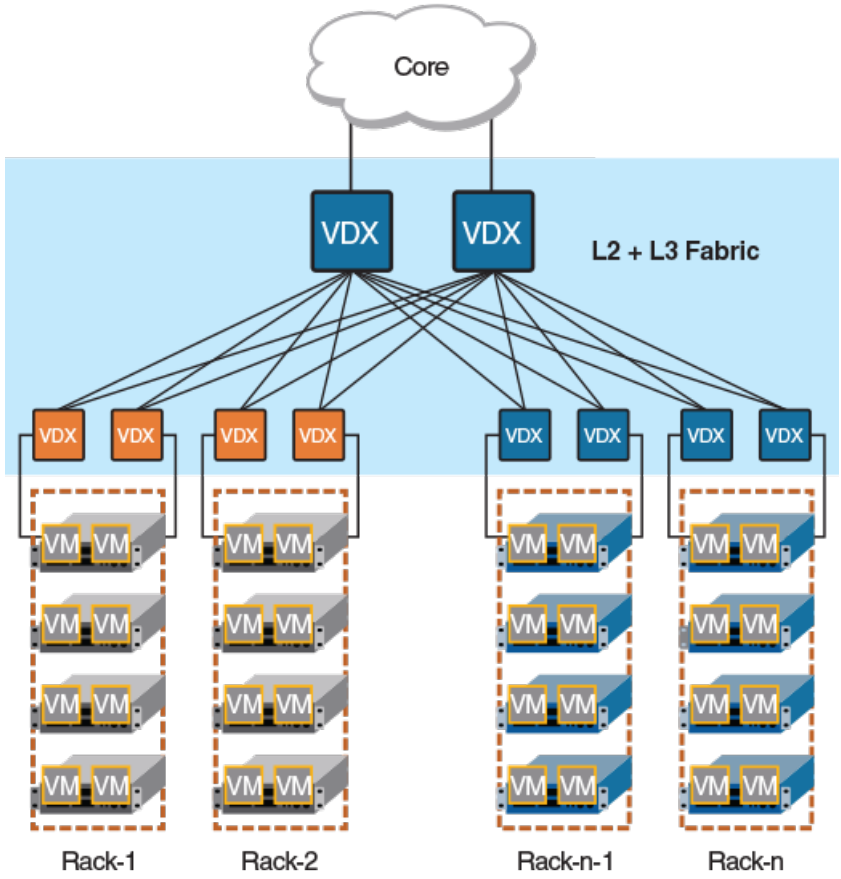
VDX 6740-based fabric

In a Brocade VDX 6740-based fabric, the gateway functionality is distributed across every VDX 6740 RBridge.

This topology is not recommended because of the following limitations:

- It requires a direct attachment between the gateway and a VXLAN-enabled rack, because the VDX-6740 cannot support TRILL-plus-VXLAN encapsulation.
- The number of server racks is limited to eight. This is constrained by the maximum number of RBridges that are allowed in a gateway.
- The VDX 6740 supports a maximum of 2000 VLANs.

FIGURE 15 VDX 6740-based fabric



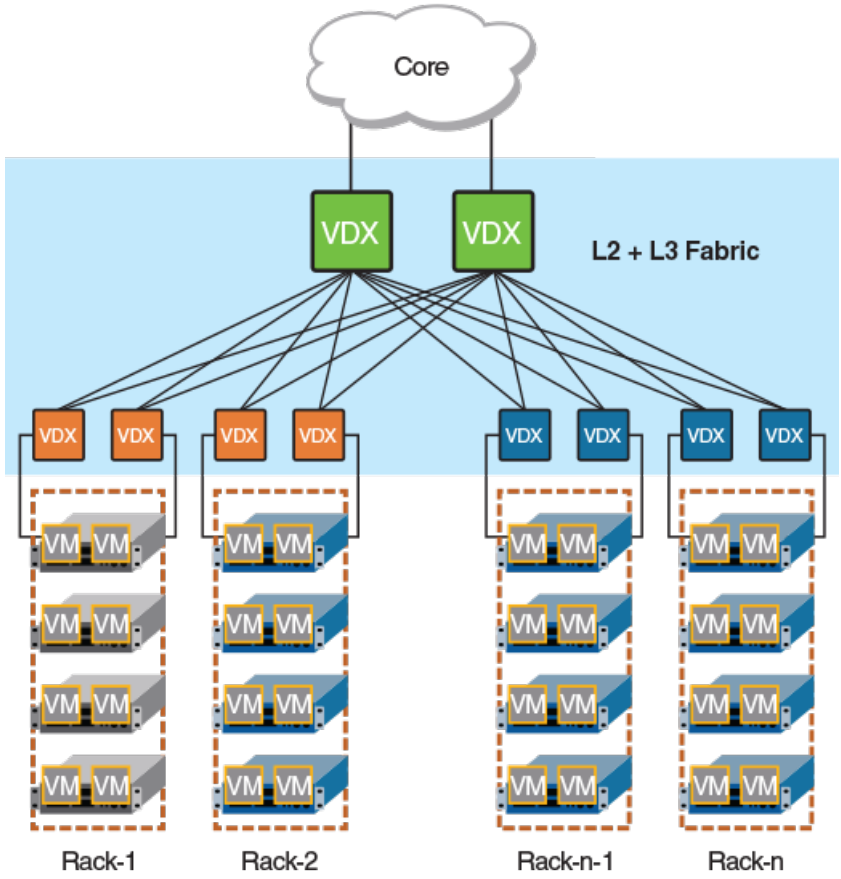
VDX 6740 and VDX 6940-based fabric

In this topology, a gateway comprises a mix of Brocade VDX 6740 and VDX 6940 RBridges.

The VDX 6740 is placed at top of rack to serve directly connected VXLAN servers. The VDX 6740 gateway handles VNI classifications for east-west traffic, thereby offloading traffic to the VDX 6940 gateway at the aggregation layer to serve other north-south traffic classifications.

Although this topology helps scale out the number of overlay networks supported in the fabric, it has the same limitations as in the previous topology, where the VDX 6740 is placed at top of rack.

FIGURE 16 VDX 6740 and VDX 6940-based fabric



Distributed VXLAN gateways RBridge scalability

Every RBridge in the fabric can act as a gateway. However, because dynamic virtual RBridge IDs (VRBs) support a maximum of eight RBridges, a tunnel VRB-ID can represent a maximum of eight RBridges out of all the gateway RBridges that are specified in the overlay-gateway configuration.

> **NOTE**
> With the current release, the maximum number of gateway RBridges deployed at the aggregation layer is four. If this number is exceeded, the RBridges that are reachable by means of VRBs is nondeterministic.

Distributed VXLAN gateways upgrade/downgrade considerations

There are no restrictions to upgrading from Network OS 5.0.x to Network OS 6.0.1. For the Brocade VDX 6740, after an upgrade or downgrade the existing four-node VDX 6740 fabric can continue to serve as a four-node gateway fabric, but it should not join another fabric. In addition, the Virtual Fabrics extension gateway at the aggregation layer can continue to act as an extension gateway, but it cannot be configured as a VXLAN gateway.

> **ATTENTION**
> In a downgrade from the current release, any new commands introduced in this release must be removed from devices before the downgrade is honored. Also, TRILL+VXLAN functionality is lost during a downgrade, and there is no warning to the user.

Distributed VXLAN gateways limitations

The following functions are not supported for distributed VXLAN gateways:

- BUM optimization
- Loop detection that involves both a tunnel and a nontunnel path
- Flow-based load balancing for tunnels over router ports
- Routing protocols over tunnels
- More than one VTEP per fabric
- QoS that is limited to DiffServ tunneling pipe mode
- A SPAN destination that is not a tunnel

Configuring a distributed VXLAN gateway

This task uses the **overlay-gateway** command and related commands to configure a distributed VXLAN gateway.

1. Enter global configuration mode.

```
device# configure terminal
device(config)#
```

2. Enter the **overlay-gateway** command and specify a gateway, entering VXLAN overlay gateway configuration mode.

```
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)#
```

3. Enter the **type** command to specify the gateway type as Layer 2 extension.

```
device(config-overlay-gw-gateway1)# type layer2-extension
```

4. Enter the **ip interface** command to specify Loopback 1 as the IPv4 interface.

```
device(config-overlay-gw-gateway1)# ip interface loopback 1
```

5. Enter the **attach rbridge-id** command to attach RBridges as appropriate for your network, as in the following example.

```
device(config-overlay-gw-gateway1)# attach rbridge-id add 3,4
```

6. Enter the **map vlan vni** command with the **auto** keyword to enable automatic VLAN-to-VNI mapping for every VLAN associated with the tunnel.

```
device(config-overlay-gw-gateway1)# map vlan vni auto
```

7. Enter the **site** command to create a remote Layer 2 extension site in a VXLAN overlay gateway context and enable VXLAN overlay gateway site configuration mode.

```
device(config-overlay-gw-gateway1)# site site1
device(config-site-site1)#
```

8. Enter the **ip address (VXLAN)** command to specify the destination IPv4 address of a tunnel.

```
device(config-site-site1)# ip address 10.1.1.1
```

9. Enter the **extend vlan** command to configure a switchport VLAN (or VLANs, as appropriate) for the tunnel to the containing site, as in the following example.

```
device(config-site-site1)# extend vlan add 1005
```

10. Enter the **activate (VXLAN gateway)** command to activate the gateway, and return to VXLAN overlay gateway configuration mode.

```
device(config-site-site1)# activate
device(config-overlay-gw-gateway1)#
```

Troubleshooting and managing distributed VXLAN gateways

A variety of options are available for managing and troubleshooting distributed VXLAN gateways.

Troubleshooting

The following examples illustrate commands used to confirm VTEP and tunnel configurations, VLAN activation, and MAC addresses, as well as to view gateway statistics.

To confirm VTEP and tunnel configurations:

```
device# show running-config overlay-gateway
 overlay-gateway gateway2
     type layer2-extension
     ip interface Ve 29 vrrp-extended-group 255
     attach rbridge-id add 1-2
     attach vlan 2
     activate

device# show tunnel brief rbridge-id 1
Tunnel 61441, mode VXLAN, rbridge-ids 10
Admin state UP, Oper state UP
Source IP 60.60.60.29, Vrf default-vrf
Destination IP 20.1.1.1

device# show tunnel 61441
Tunnel 61441, type nsx, rbridge-ids 1,2
Admin state UP, Oper state UP
Source IP 60.60.60.29, Vrf default-vrf
Destination IP 60.60.60.184

device# show tunnel statistics
Tnl ID   RX packets      TX packets      RX bytes        TX bytes
======== =============== =============== =============== ================
61441    123             456             (NA)            4560
61442    567             890             (NA)            8900

device# show system internal tnlmgr tunnel all
[is this for public consumption??]
```

To confirm VLAN activation:

```
device# show vlan brief
Total Number of VLANs configured    : 14
Total Number of VLANs provisioned   : 14
Total Number of VLANs unprovisioned : 0
VLAN             Name            State                     Ports           Classification
(F)-FCoE                                                   (u)-Untagged
(R)-RSPAN                                                  (c)-Converged
(T)-TRANSPARENT                                            (t)-Tagged
================ =============== ========================= =============== ====================
1                default         ACTIVE                    Po 333(t)
                                                           Po 444(t)
2                 VLAN2          ACTIVE                    Tu 61441(t)     vni 3029)

1002(F)          VLAN1002        INACTIVE(no member port)
```

To confirm MAC addresses:

```
device# show mac-address-table
VlanId   Mac-address       Type      State        Ports
2        0000.0000.0001    Dynamic   Active       Tu 61441
2        0000.0000.0002    Dynamic   Active       Tu 61441
2        0000.0000.0003    Dynamic   Active       Tu 61441
2        0000.0000.0004    Dynamic   Active       Tu 61441
2        0000.0000.0005    Dynamic   Active       Tu 61441
2        0000.0000.0006    Dynamic   Active       Tu 61441
20       0027.f880.1890    System    Remote       XX 40/X/X
20       0027.f880.328f    Dynamic   Active       Po 1
```

To view gateway statistics:

```
device# show overlay-gateway name test vlan statistics
VLAN ID   RX packets           TX packets
========= ==================== ====================
30        1234                 5678
```

Enabling SPAN

You can use the Switched Port Analyzer (SPAN) protocol to monitor traffic by means of the **monitor session** command, as in the following examples:

```
device(config-overlay-gw-gsteway3)# monitor session 1 direction both remote-endpoint 1.2.3.4 vlan add 2

device(config-overlay-gw-gateway3)# monitor session 1 direction both remote-endpoint any vlan add 2
```

NOTE

A SPAN destination must first be created outside the overlay gateway (as a monitor session). All regular SPAN sessions are supported. For details, see Switched Port Analyzer on page 269.

Virtual Fabrics

Virtual Fabrics overview

The Virtual Fabrics feature delivers Layer 2 Multitenancy solutions that provide support for overlapping VLANs, VLAN scaling, and transparent VLAN services by providing both traditional VLAN service and a transport service. These services are offered by provisioning a Virtual Fabric (VF) in the data center. A VF operates like a regular 802.1Q VLAN, but has a 24-bit address space that allows the number of networks to scale beyond the standard 4K (4096) limit. The transport service is provided by configuring a transport VF, whereas traditional Layer 2/Layer 3 VLAN service is provided by configuring a service VF.

The Virtual Fabrics feature is deployed in data centers where logical switch partitioning and server virtualization require a large number of customer VLAN domains that must be isolated from each other in the data plane. On the hardware platforms that support this feature, the Brocade VDX 8770 series and the Brocade VDX 6740 series running Network OS release 4.1.0 or later, the VLAN ID range is extended from the standard 802.1Q limit of 4095, to extend through 8191 on both a local RBridge and in a single VCS Fabric.

Data center virtualization, such as that provided by VMware vCenters, challenges network design in a variety of areas. Large numbers of networks can be required to support the virtualization of server hosts and multiple-tenant virtual machines (VMs), with Ports on Demand (POD) for the virtual data center (vDC) leading to POD configurations replicated at different ports. Such virtualization topologies are inherently largely independent of physical networks, with VM network configurations decoupled from the addressing and configurations of physical networks. The underlying Layer 2/Layer 3 infrastructure is an extension of a VMware virtual switch, or vSwitch, requiring address mapping at the boundary between the physical and virtual networks. In addition, the requirement for VM mobility makes it necessary to support the migration of VMs across the Virtual Cluster Switching (VCS) data center or across geographically separated sites, independently of the connectivity of the underlying infrastructure.

When a fabric is VF-enabled, existing VLAN configurations can apply to any VLAN. When a fabric is not VF-enabled, VLAN classification is not supported, and VLAN configurations are 802.1Q VLANs with VLAN IDs 1 through 4095.

A Virtual Fabric is just like a regular 802.1Q VLAN, but with a 24-bit address space that has the potential to support up to approximately16 million VLANs to be provisioned in the fabric. This VF VLAN address space is common to regular 802.1Q VLANs and classified VLANs. VLAN IDs from 1 through 4095 identify a conventional 802.1Q VLAN. VLAN IDs greater than or equal to 4096, up through 8191, identify VFs that need frame classification. A VF VLAN ID is unique within a local VCS Fabric, but may not be unique across multiple VCS Fabrics.

> NOTE
> A service VF is defined on the basis of the encapsulation classification of the ingress frame, with frames classified at the edge port according to the 802.1Q VLAN ID or MAC address. For the same service VF, the 802.1Q classification rule at each interface is a link-local configuration; the rule may be different at each interface.

A service VF thus represents a virtualized, normalized VLAN domain, where different link-protocol VLAN identifiers (port number, MAC address, and customer VLAN ID, or C-VID) are mapped to the same VLAN. In other words, VMs on the same service VF belong to the same forwarding domain, even though the attachment interfaces use different classification rules. When a VM moves among these interfaces, the Layer 2 forwarding domain does not change.

Extending a service VF among VCS data centers makes it possible to migrate VMs across those data centers.

This chapter also presents an overview of the role of the Brocade VDX 6940 in supporting *distributed VXLAN gateways*, made available in Network OS 6.0.1. ""Distributed" means that the gateway can be deployed anywhere in the VCS Fabric and coexist with other non-gateway RBridges, subject to the topology constraints and other limitations as noted in the distributed VXLAN gateways section. Only the VDX 6940 supports this capability, and no special configuration is required. Using Fabric-Virtual-Gateway is one of many ways to configure the gateway IP address.

NOTE
Only Layer 2 extension gateways are supported. NSX Controller gateways are not supported.

Virtual Fabrics features

The following VLAN switch-port configurations are supported:

- Regular 802.1Q configuration (VLAN IDs 1 through 4095, with the exception of reserved VLANs)
- VLAN classification by means of a 802.1Q tag at the trunk port
- VLAN classification by means of a source MAC address at the access port

The following standard VLAN features remain supported by the service VF feature:

- Private VLANs (PVLANs)
- Layer 3 virtual Ethernet (VE) interfaces
- VLAN classifiers
- IGMP snooping
- VLAN ACLs
- Automatic Migration of Port Profiles (AMPP)
- RSPAN
- xSTP

In addition, support is now provided for transport service VFs, enabling a provisioning model in which a specific group of 802.1Q VLANs at an interface is classified into a common forwarding domain.

The maximum number of VLANs supported in this release, 802.1Q and classified, is as follows:

- There is fabric-wide support for 8K VF instances. The number of VFs supported on a local switch is platform-dependent.
- In a pure transport service deployment, port-based transport VFs are supported on every edge port, with up to 2048 VLANs (both 802.1Q and classified) across all ports.

For additional scalability details, refer to Virtual Fabrics performance considerations on page 112.

For example topologies and detailed discussion, refer to Virtual Fabrics configuration overview on page 112.

NOTE
Network OS 5.0.0 added support for conversational MAC learning (CML) on classified VLANs as well as on 802.1Q VLANs. For an overview and applicable configuration examples, refer to Conversational MAC learning on page 79. As noted above, Network OS6.0.1 added support for distributed VXLAN gateways on the Brocade VDX 6940 series for Layer 2 extension only.

Virtual Fabrics considerations and limitations

FGL limitations

Support is provided for pre-IETF standard Fine-Grained Labeling (FGLs) on ISLs. IETF defines the Ethertype for inner and outer labels as 0x893B. The outer and inner Ethertypes on ISLs are set to 0x893B and 0x8100, respectively.

FCoE VLAN limitations

FCoE VLANs can be only 802.1Q VLANs. All tenant FCoE traffic rides on the same default FCoE VLAN, 1002. This is the same as in previous releases. An FCoE VLAN can be used as a classification tag on a port if it is not configured as an FCoE port.

STP support

The correct configuration of xSTP is the responsibility of the user. Much as the user must ensure that VLAN configurations and VLAN instance mappings are consistent on all switch ports, so also the user must understand whether a specific protocol, whether RSTP, MSTP, or PVST, is applicable to the underlying physical topology when 802.1Q VLANs and VFs coexist in the fabric.

Brocade VDX 6740 series limitations

The Brocade VDX 6740 series platforms do not support full port-based C-TAG translation. Whenever there is a translation conflict among the ports because the same C-TAG is used for different VLANs, the conflict cannot be resolved without incurring severe internal VLAN ID (IVID) scalability constraints. Optimal scalability is possible only when overlapping C-TAG classification occurs across port groups.

In a pure service VF deployment, these platforms support up to 4096 802.1Q VLANs and 2048 classified VLANs, assuming that each VLAN configured on a single port constitutes a single conflict.

In a pure transport VF deployment, these platforms support port-based transport VFs on every edge port, and a total of 2048 combined 802.1Q and classified VLANs, assuming that each VLAN is configured on a single port.

VCS extension through VXLAN

Network OS 5.0.0 and later supports the use of VFs by means of VXLAN Layer 2 extension, to extend a VCS Fabric to another VCS Fabric.

Distributed VXLAN gateways limitations

Refer to the "Distributed VXLAN gateways" section for supported topologies and other details.

Distributed VXLAN gateways overview

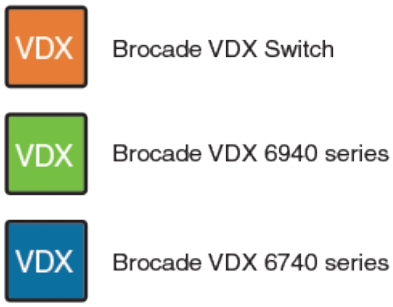
The distributed VXLAN gateways feature eliminates the need for an external gateway device.

Prior to Network OS 6.0.1, VXLAN gateways had to be connected to the VCS Fabric (for example, a four-node gateway fabric) as an external device in order to bridge the overlay network and the physical networks that are interconnected by the VCS Fabric. Beginning with the current release, the gateway function can be hosted by the VCS RBridges. This eliminates the need for an external gateway device and optimizes network resources, improving network performance in the data center.

> NOTE
> Existing VRRP-E implementations cannot support more than four RBridges in a session. As a result, VRRP-E-based VXLAN gateways are also limited to a maximum of four RBridges.

The following sections present various use cases, both supported and unsupported, and their corresponding topologies. Refer to the following legend for those topologies.

FIGURE 17 Distributed VXLAN gateways legend



NOTE

In the initial release, this feature supported only Virtual Fabrics Extension deployments. Beginning with Network OS 7.0.1, support is provided for NSX Controller deployments.

Distributed VXLAN gateways supported topologies

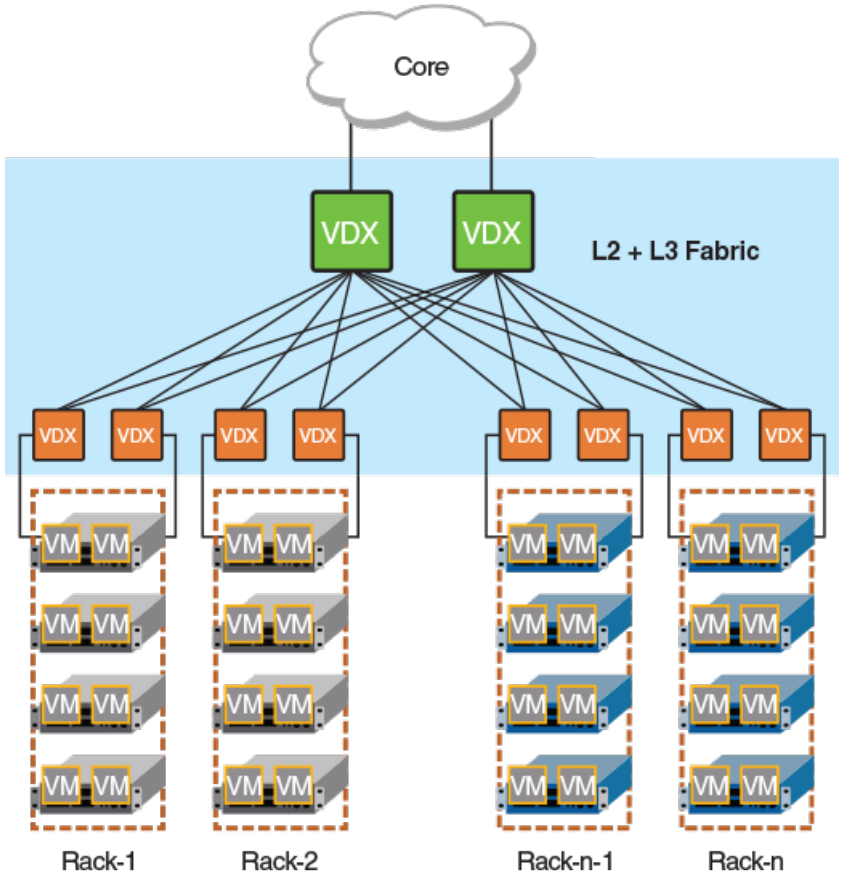
The following sections present the topologies that are supported in this release.

VDX 6940 at the aggregation layer

A Brocade VDX 6940 at the aggregation layer provides gateway functions for a VXLAN hypervisor at top of rack (ToR) or in the core.

A distributed VXLAN gateway makes it possible to connect to a hypervisor through a TRILL fabric, as the Brocade VDX 6940 supports TRILL-plus-VXLAN encapsulation. Refer to the following figure.

FIGURE 18 VDX 6940 in a Layer 2/Layer 3 fabric



In this topology, bridging a physical server and VXLAN servers that are located in the same rack requires interaction with an aggregation gateway, introducing an extra hop. The gateway supports VXLAN Network Identifier (VNI) classification for both east-west and north-south traffic. Note the following considerations:

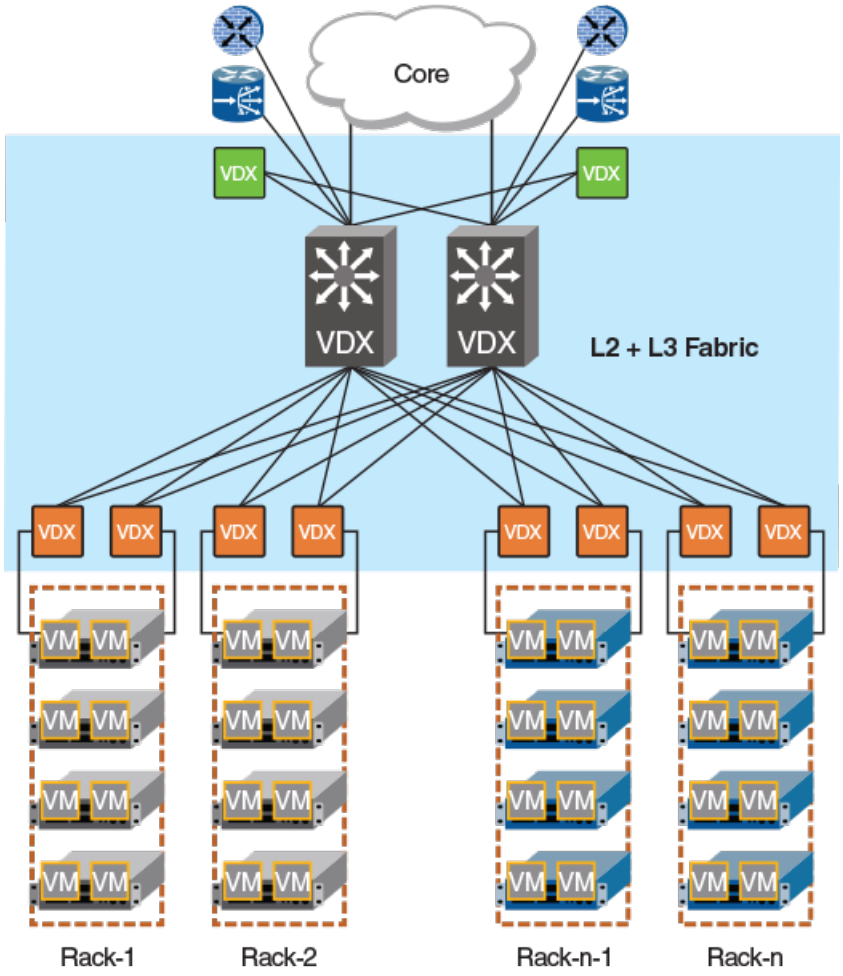
- The number of overlay networks that are supported in the fabric is limited by ASIC resources for the VNI classifications.
- The Brocade VDX 6940 is not supported at top of rack. This prevents VLAN-to-VXLAN traffic from "tromboning" in case the ToR gateway that does the VXLAN encapsulation is not in the same rack that holds the VXLAN server.

VDX 6940 as an appliance

A Brocade 6940 gateway can be inserted into the fabric as an appliance.

The following figure illustrates how a data center can use a VXLAN-incapable aggregation switch (such as a Brocade VDX 8770) to provide connectivity to the core while the switch is attached to a VXLAN gateway functioning as an appliance. Traffic between the physical server and a VXLAN-enabled server must always take an extra hop through the VDX to reach the appliance gateway.

FIGURE 19 VDX 6940 as a gateway appliance



Distributed VXLAN gateways unsupported topologies

The following topologies, although they can provide a certain degree of functionality, are not supported by Brocade for this feature, for reasons detailed in the sections below.

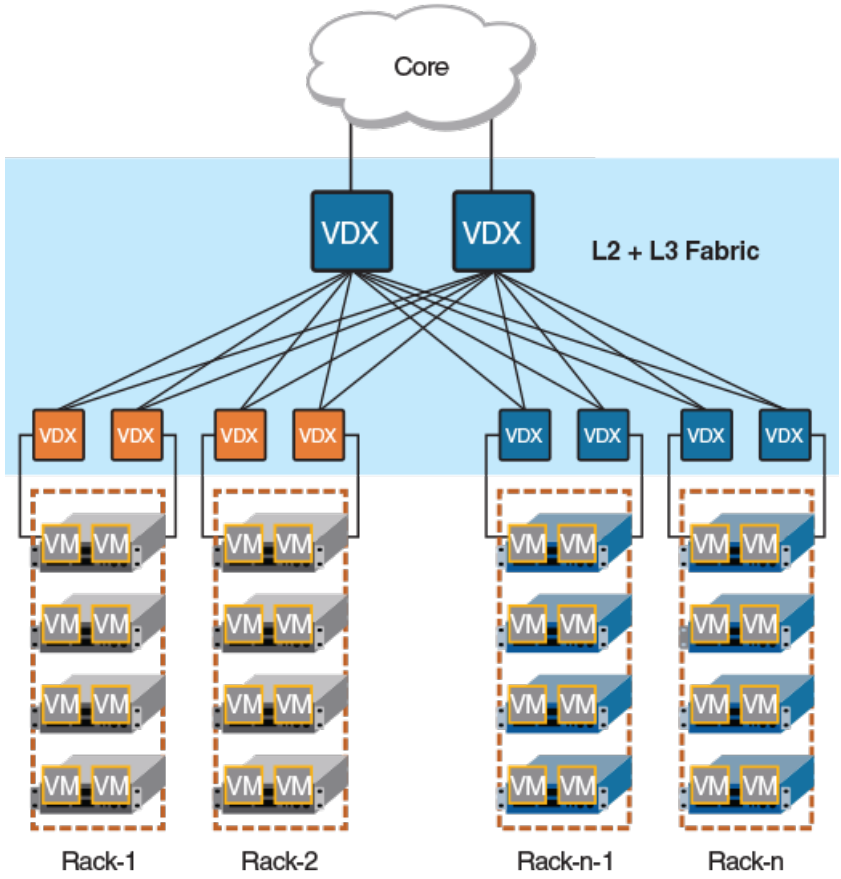
VDX 6740-based fabric

In a Brocade VDX 6740-based fabric, the gateway functionality is distributed across every VDX 6740 RBridge.

This topology is not recommended because of the following limitations:

- It requires a direct attachment between the gateway and a VXLAN-enabled rack, because the VDX-6740 cannot support TRILL-plus-VXLAN encapsulation.
- The number of server racks is limited to eight. This is constrained by the maximum number of RBridges that are allowed in a gateway.
- The VDX 6740 supports a maximum of 2000 VLANs.

FIGURE 20 VDX 6740-based fabric



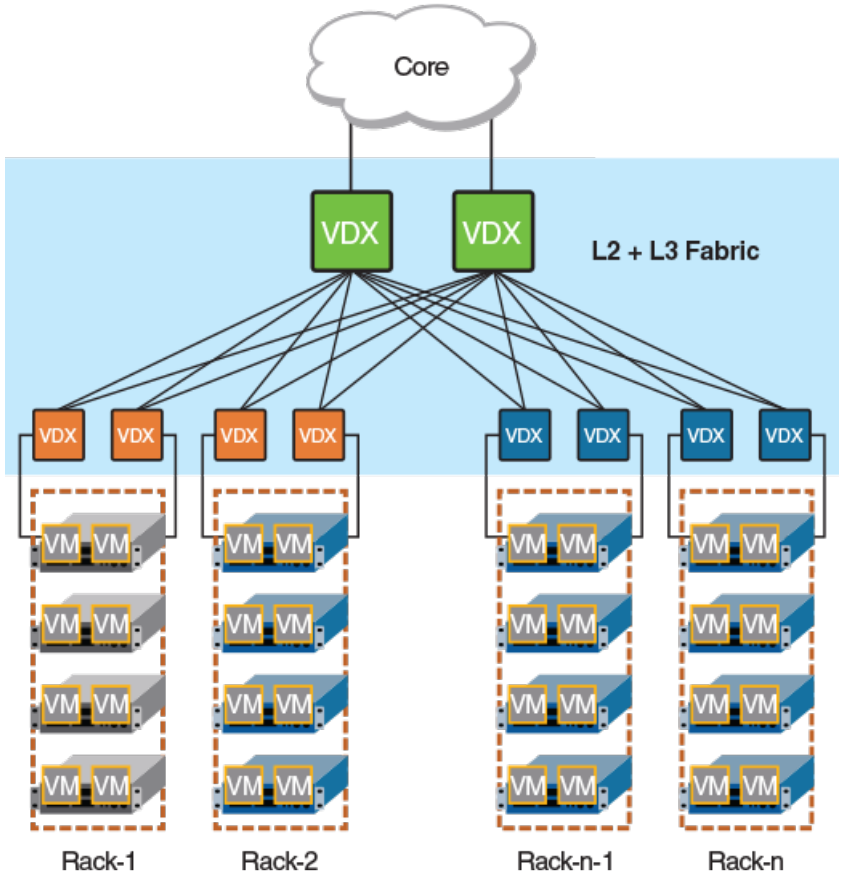
VDX 6740 and VDX 6940-based fabric

In this topology, a gateway comprises a mix of Brocade VDX 6740 and VDX 6940 RBridges.

The VDX 6740 is placed at top of rack to serve directly connected VXLAN servers. The VDX 6740 gateway handles VNI classifications for east-west traffic, thereby offloading traffic to the VDX 6940 gateway at the aggregation layer to serve other north-south traffic classifications.

Although this topology helps scale out the number of overlay networks supported in the fabric, it has the same limitations as in the previous topology, where the VDX 6740 is placed at top of rack.

FIGURE 21 VDX 6740 and VDX 6940-based fabric



Distributed VXLAN gateways RBridge scalability

Every RBridge in the fabric can act as a gateway. However, because dynamic virtual RBridge IDs (VRBs) support a maximum of eight RBridges, a tunnel VRB-ID can represent a maximum of eight RBridges out of all the gateway RBridges that are specified in the overlay-gateway configuration.

NOTE

With the current release, the maximum number of gateway RBridges deployed at the aggregation layer is four. If this number is exceeded, the RBridges that are reachable by means of VRBs is nondeterministic.

Distributed VXLAN gateways upgrade/downgrade considerations

There are no restrictions to upgrading from Network OS 5.0.x to Network OS 6.0.1. For the Brocade VDX 6740, after an upgrade or downgrade the existing four-node VDX 6740 fabric can continue to serve as a four-node gateway fabric, but it should not join another fabric. In addition, the Virtual Fabrics extension gateway at the aggregation layer can continue to act as an extension gateway, but it cannot be configured as a VXLAN gateway.

ATTENTION

In a downgrade from the current release, any new commands introduced in this release must be removed from devices before the downgrade is honored. Also, TRILL+VXLAN functionality is lost during a downgrade, and there is no warning to the user.

Distributed VXLAN gateways limitations

The following functions are not supported for distributed VXLAN gateways:

- BUM optimization
- Loop detection that involves both a tunnel and a nontunnel path
- Flow-based load balancing for tunnels over router ports
- Routing protocols over tunnels
- More than one VTEP per fabric
- QoS that is limited to DiffServ tunneling pipe mode
- A SPAN destination that is not a tunnel

Virtual Fabrics upgrade and downgrade considerations

The following Virtual Fabric upgrade and downgrade considerations need to be kept in mind.

Virtual Fabrics backward compatibility

In Network OS release 4.1.0, the Virtual Fabrics feature is capable of support both regular 802.1Q and VF configurations. RBridges that upgrade to Network OS 4.1.0 will continue to support 802.1Q configurations that exist in the fabric.

Virtual Fabrics forward compatibility

The Virtual Fabrics feature is supported on Brocade VDX 6740 series, and VDX 8770 series platforms beginning with Network OS 4.1.0. However, whether VF configuration is permitted on these platforms is a global fabric-wide decision, and is not a platform-based decision. VF configuration is permitted only when a fabric consists of VF-capable RBridges running in logical chassis cluster mode.

In earlier releases, every 802.1Q VLAN had to be configured on every RBridge. This configuration had to be applied to every RBridge, even when that RBridge serves as a transit and did not terminate the VLAN. This is because the RBridge only forwarded frames on an Inter-Switch Link (ISL) that were associated with a configured VLAN. Beginning with Network OS 4.1.0, this restriction does not apply. In logical chassis cluster mode, VLAN configurations are automatically distributed to all nodes; the user does not have to configure VLANs on transit RBridges. A transit RBridge can always forward VLAN frames that arrive on an ISL. If the frame exists in the fabric, it must have been allowed to enter the fabric at the edge. In fabric cluster mode, Network OS 4.0.0 configuration rules still apply; the user must configure VLANs on every RBridge.

NOTE

Also refer to "Distributed VXLAN gateways upgrade/downgrade considerations" above.

Virtual Fabrics operations

This section summarizes the fundamental behavior of Virtual Fabrics.

Enabling and disabling a Virtual Fabric

Virtual Fabrics can be enabled only in logical chassis cluster mode. In a VF-incapable fabric, ISL encapsulation is based on C-TAGs. In a VF-enabled fabric, ISL encapsulation is based on Fine-Grain Labels (FGLs), using both C-TAGs and S-TAGs. The VF is enabled only when the user issues the **vcs virtual-fabric enable** command. This enables the transition from one encapsulation type to another without disrupting existing traffic.

Enabling VFs

To ensure that there is no data disruption, the ISL encapsulation transition must be coordinated at the fabric level without toggling the link state. When the fabric is in a VF-capable state, the user executes the fabric-wide **vcs virtual-fabric enable** command to enable this transition. The command is distributed to all RBridges in the fabric and is executed in multiple stages across nodes.

> NOTE
> If the fabric state is VF-incapable, the **vcs virtual-fabric enable** command will not succeed.

Disabling VFs

To disable VFs in the fabric, the user must first remove all VF configurations in the fabric before issuing the **no vcs virtual-fabric enable** command. This command is distributed to all RBridges in the fabric. Each RBridge reverses the stage execution from what was done to enable the VF. When ISL encapsulation returns to being C-TAG based, the fabric is in a VF-capable state.

> NOTE
> Prior to issuing the **no vcs virtual-fabric enable** command, the user must ensure that all new commands and enhanced commands that reference VFs (VFs with IDs greater than 4095) in the fabric are removed from the running configuration. Otherwise, the command will be rejected.

Joining a switch to the fabric

If VFs are disabled in the startup configuration, the RBridge will not able to join a VF-enabled fabric upon reboot. Although the RBridge may form ISLs with a neighbor RBridge at the link level, the RBridge will be segmented from the VF-enabled cluster because of a capability mismatch.

In order for the RBridge to join a VF-enabled cluster, the procedures described in the "Upgrading and downgrading firmware with Virtual Fabrics" task must apply to the segmented RBridge. At the end of the procedures, the segmented RBridge will also be in the VF-enabled state. All segmented ISLs on the RBridge are automatically toggled so that it can rejoin the fabric.

Default Virtual Fabrics state

A switch after a **netinstall** is VF-incapable in the default configuration. It will fail to join a VF-enabled cluster when it reboots.

The default VF state in the startup configuration is changed when the **copy default-config startup-config** command is executed. The resulting state is the same as the current VF state of the fabric.

Virtual Fabrics configuration overview

This section addresses a variety of Virtual Fabrics architecture scenarios and configuration issues related to service VFs.

Virtual Fabrics performance considerations

VLAN configurations, whether for 802.1Q or classified VLANs, are VCS Fabric global configurations that are distributed to all RBridges in the fabric. If there are 100K VLANs in the data center, this implies that as RBridge will receive all those configuration, even though the RBridge could support a maximum of 8K (8192) VLANs. The processing of VLAN configurations for classified VLANs that are not actually provisioned in the RBridge introduces performance overhead. The impact of this will be manifest in configuration playback during system boot up and will result in a longer time for the fabric to reach a ready state to forward data.

Feature scalability

The scalability numbers of VLAN features remains same as in the previous release. The following lists VF resource numbers for the Brocade VDX 8770, VDX 6740, and VDX 6940 series platforms.

TABLE 9 VF resource numbers for Brocade VDX 8770, VDX 6740, and VDX 6940 series

Resources	VDX 8770 series	VDX 6740 series	VDX 6940 series
802.1Q VLANs per switch	4096	4096	4096
VFs per switch	4096	2000	4096
VLANs per fabric	8192	8192	8192
MAC-based classifications per switch	2048	256	1000
Service VF with overlapping C-TAG (best case)	4096	2000	4096
Service VF with overlapping C-TAG (worst case)	4096	64	4096
AMPP port profiles	1000	1000	1000
Transport VFs (best case)	1000	1000	1000
Transport VFs (worst case)	1000	178	144

Maximum number of VLANs

The target number of VLANs supported is 8K at a local RBridge. These may be traditional 802.1Q VLANs or service VFs. Below are some factors that determine the actual number of VLAN IDs that are configurable on a local RBridge constrained by the availability of ASIC resources to support internal VLAN IDs (IVIDs).

AMPP port-profile provisioning

When a port becomes a profile-port, IVIDs are allocated internally for VLANs that are defined in a port-profile domain associated with the interface or VLANs that are defined in the port-profiles associated with the interface. These resources are consumed before the VLANs are provisioned, underutilizing the IVID resources if a VM does not appear on these VLANs. The overbooked IVIDs will not be available for VLANs configured by the **switchport** command.

Number of VE interfaces

The **interface ve** command creates a VE interface on a specified VLAN on a specified RBridge. Because the VLAN flood domain must extend within the VCS Fabric by means of ISLs for routing purposes, an IVID must be allocated for the VLAN even though the VLAN is not configured on any local RBridge switch ports, imposing a scalability issue. Each VE interface consumes an IVID from the resource pool. The total number of interfaces (without switch ports) and VLANs that are provisioned cannot exceed 8192 on the VDX 8770 series and VDX 6740 series.

Number of VLAN ACLs

A VLAN ACL requires an IVID allocation for the target VLAN. If the target VLAN is configured on the local switch port, the ACL can be applied on the IVID for this VLAN. However, on the switch where the VLAN is transiting (that is, it is not configured on any switch port), an IVID must still be allocated for the ACL entry. The maximum number of VLANs that can be configured on the switch is determined by the maximum number of IVIDs minus the number of transit VLAN ACLs that are configured on the switch.

Brocade VDX 6740 series limitations

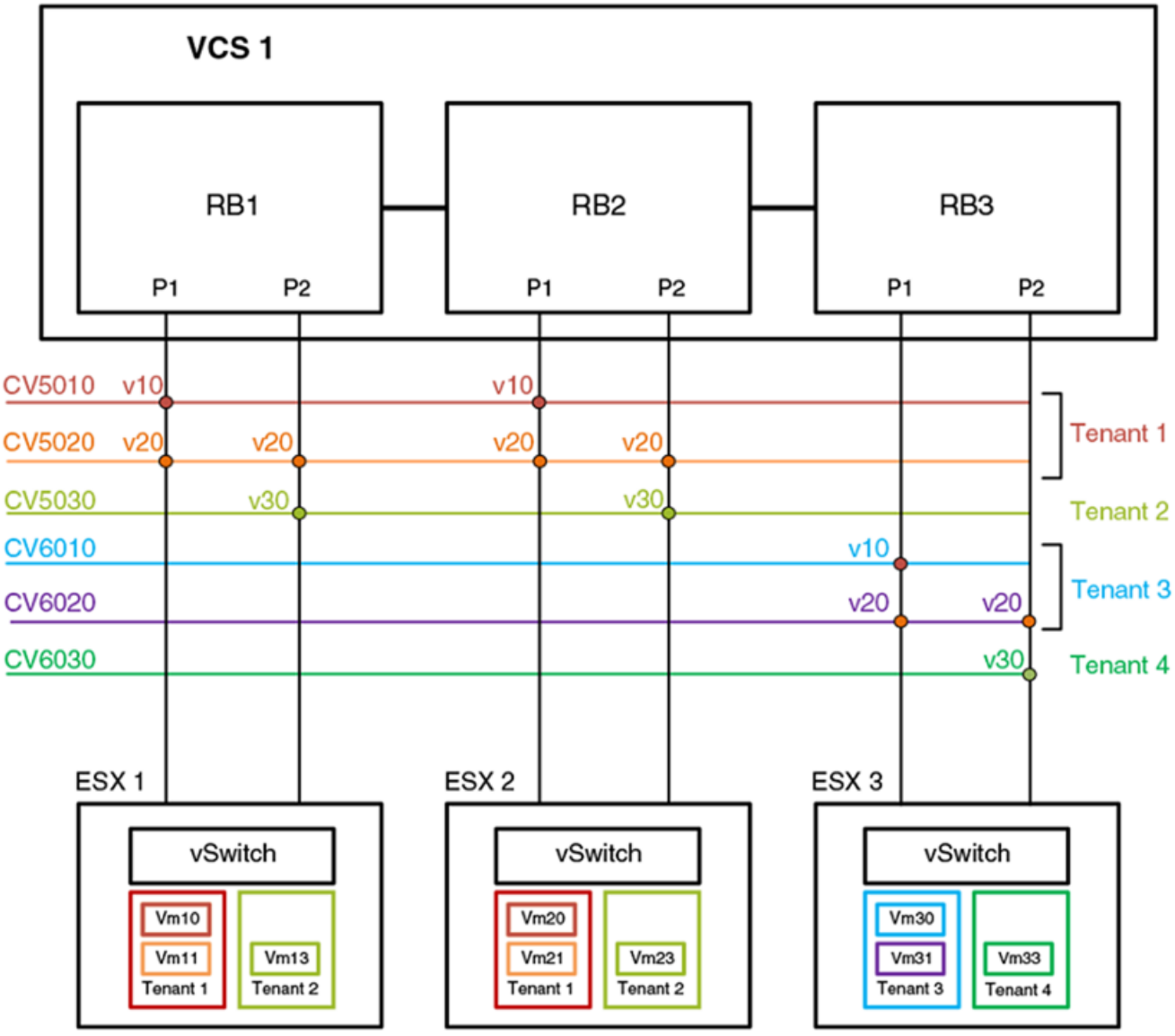
The Brocade VDX 6740 series RBridge supports a total of 6096 (4096 + 2000) VLANs, 4096 of which must be 802.1Q VLANs. The other 2000 VLANs can be configured as service VFs, assuming that the VLAN is configured on a single port. This limitation comes from the VPN table size on this platform.

VLAN virtualization

When a cloud computing provider provisions a virtual datacenter by replicating server-rack ports on demand (PODs) across server ports, different tenant domains exist but with overlapping 802.1Q VLANs at the server ports. The tenant domains are isolated by mapping the 802.1Q VLAN at each interface into a different VLAN forwarding domain. This capability allows the switch to support more than the 4K VLANs permitted by the 802.1Q address space.

In the example VMware topology shown in the following figure, the data center has three PODs, provided by RBridges RB1, RB2, and RB3. All three PODs (VMware ESXi hypervisors 1 through 3) have an identical pre-installed configuration. Each POD supports two tenants. The first tenant can have two applications running on VFs 10 and 20. The other tenant has only one application, running on VF 30. Here, four tenant applications are provisioned. Tenant 1 and 2 applications run on ESX1 and ESX2. Tenant 3 and 4 applications run on ESX3.

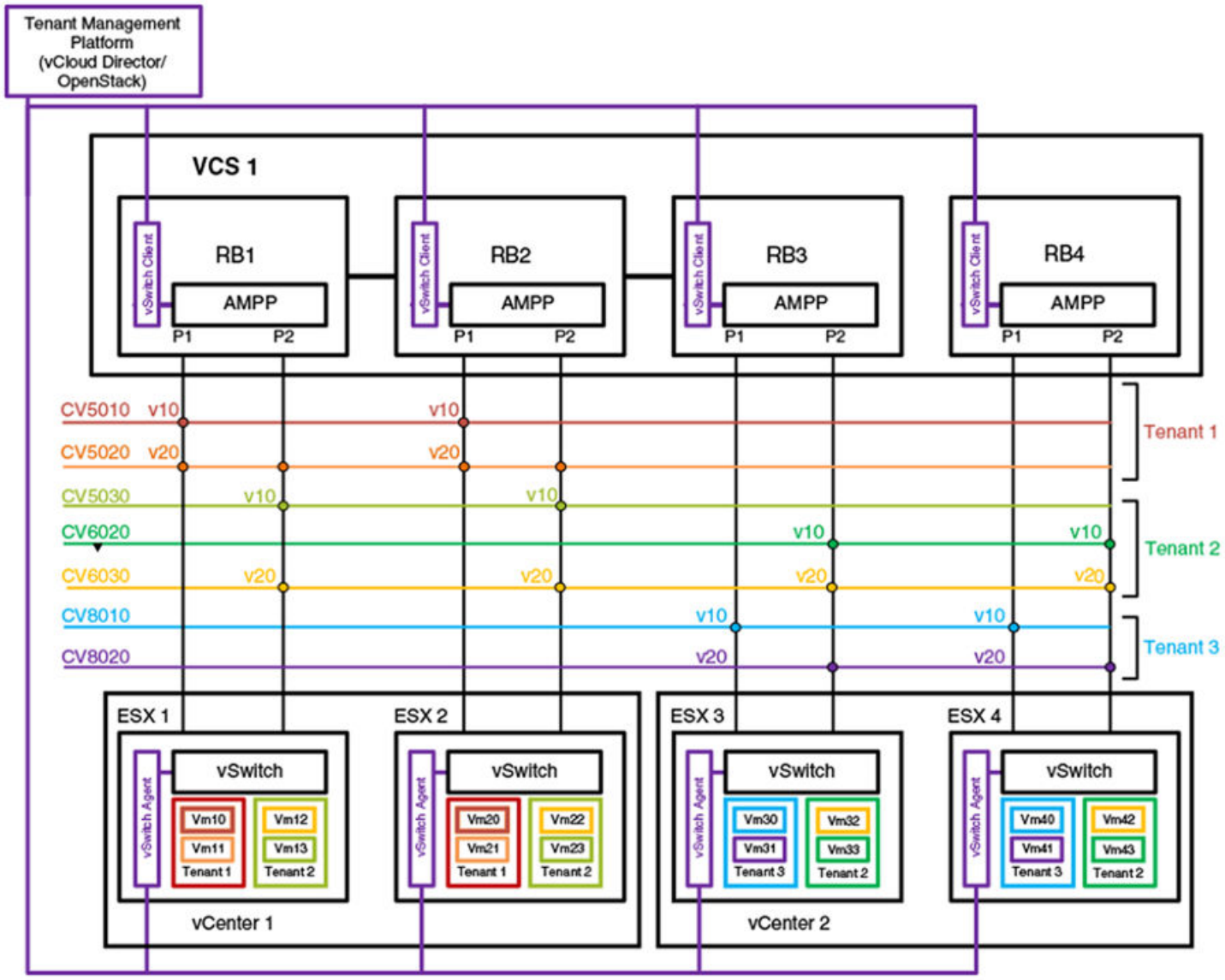
FIGURE 22 VLAN virtualization



Virtual data center deployment

The following figure illustrates an example VDC infrastructure that supports a VMware deployment.

FIGURE 23 VDC infrastructure



In a VMware-based cloud provider network, a VCS Fabric is connected to multiple vCenters, where each data center manages its own set of tenant networks. VMware vCloud/OpenStack is responsible for orchestrating tenant VLAN configuration through the vCenter agent integrated into a VCS RBridge and its ESXi servers. Each data center connects to the VCS Fabric by means of dedicated edge ports. The ability of the VCS Fabric to support 802.1Q VLAN virtualization allows each data center to support more than 4000 tenant VLANs. ESXi servers may use the same 802.1Q VLAN to represent different tenant VLANs at the edge port, or have them belong to a

single VLAN domain. A vCenter agent running at an RBridge achieves VLAN virtualization by collating information obtained from the ESXi servers and the vCenter database. AMPP port profiles with service VF classifications are configured on the respective server ports.

The topology in the figure shows two vCenters. vCenter1 is connected to VCS RBridges1 and2. Because the VCS Fabric supports VLAN virtualization, the vCenter can assign two tenant networks, CV5010 and CV5030, that use the same 802.1Q VLAN (VLAN 10) on the ESXi server. Similarly, in vCenter2, three tenant VLANs —CV5030, CV6020, and CV8010— are configured, each representing a unique VLAN domain, but all using the same customer classification, C-TAG 10. If a VM application needs to run across applications, then the same service VF can be configured on both vCenters; this is illustrated by CV6030, which is configured at all RBridges and uses the same C-TAG (C-TAG 20).

The service VF configuration at each edge port can be done as part of an AMPP configuration or automatically through vCenter orchestration. (Refer to VMware documentation for details of the vCenter Orchestrator.)

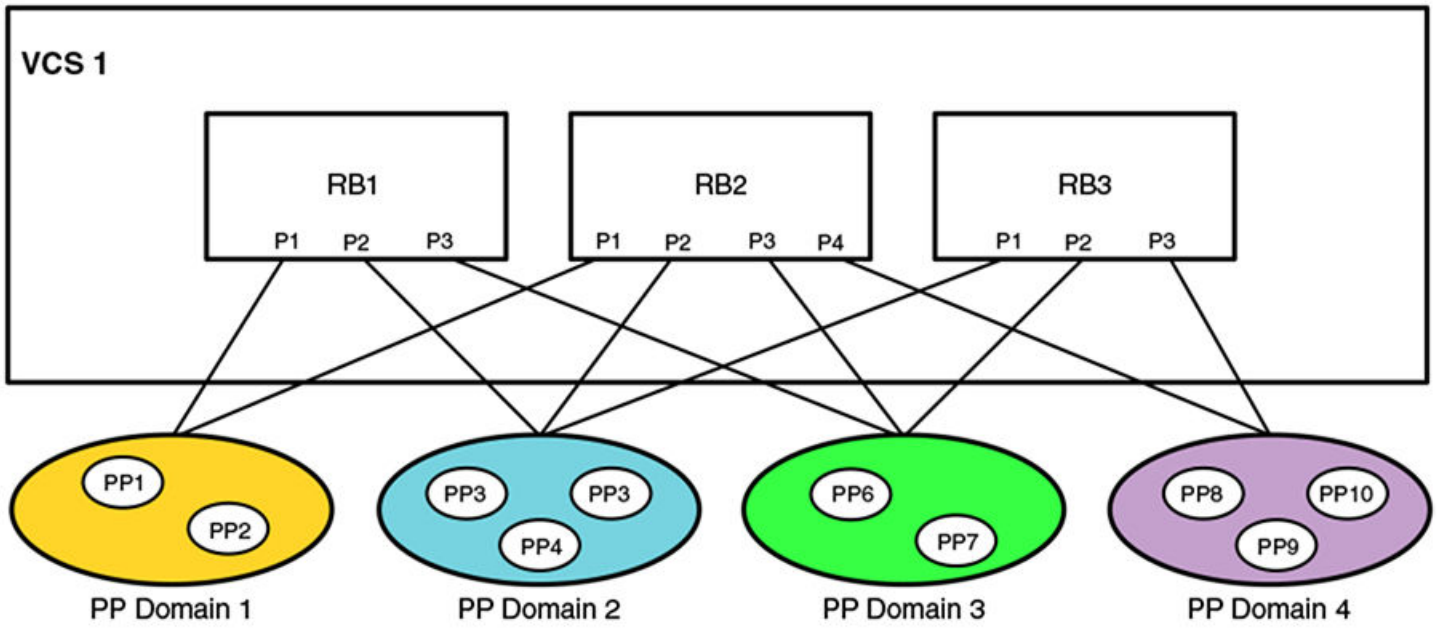
AMPP provisioning with service VFs

When the Automatic Migration of Port Profiles (AMPP) feature is used in Network OS 4.1.0 and later, a VCS Fabric is partitioned into port-profile (PP) domains. A PP domain is a set of port-profiles whose service VF ID cannot have conflicting C-TAG or MAC classifications. A port-profile domain supports a maximum of 4096 service VFs. The scope of VM mobility is defined by the set of interfaces onto which the port-profile domain is applied.

Port-profile domain topology

The following illustrates an example VCS Fabric that serves four port-profile domains across three RBridges.

FIGURE 24 Port-profile domains



Port-profile configuration rules

The VLAN classifications in a port-profile domain follow these rules:

1. Each profiled-port is associated with a port-profile domain. Different profiled-ports (on the same RBridge) may belong to different domains.
2. The same port-profile may be associated with multiple domains.
    a. This allows a service VF that is defined in one domain to extend into another domain.
    b. The VM MAC address must be unique within a domain.
3. The service VF classification rules within a domain are as follows:
    a. MAC-based classification is allowed only on an access port.
    b. C-TAG classification is allowed only on a trunk port
    c. Service VF classification cannot overlap across port-profiles in the same port-profile domain.
    d. Classification can overlap across PP domains.
4. The following example configurations are allowed for classification across all port-profiles in the same domain:
    a. **switchport trunk allow vlan add 5000 ctag 10**
    b. **switchport trunk allow vlan add 6000 ctag 20**
    c. **switchport trunk allow vlan add 7000 ctag 30**
5. The following example configurations are allowed for MAC-based classification on an access port:
    a. **switchport access vlan 8001**
    b. **switchport access vlan 8002 mac 2.2.2**
    c. **switchport access vlan 8002 mac 3.3.3**
6. The following example configurations are disallowed across all profiles in the same port-profile domain.
    a. Overlapping VF VLANs
        - **switchport trunk allow vlan add 8000 ctag 80**
        - **switchport trunk allow vlan add 8000 ctag 800**
    b. Overlapping C-TAG:
        - **switchport trunk allow vlan add 5000 ctag 10**
        - **switchport trunk allow vlan add 6000 ctag 10**
    c. Overlapping MAC address across all profiles in the same domain:
        - **switchport access vlan 8002 mac 2.2.2**
        - **switchport access vlan 8003 mac 2.2.2**
    d. Conflicting access and trunk service VF configurations:
        - **switchport access vlan 8000 mac 2.2.2**
        - **switchport trunk vlan add 8000 ctag 20**
7. The following configuration can be present on only one port-profile domain; it cannot coexist with other C-TAG-based classifications in that domain:
    a. **switchport trunk allow vlan all**
    b. When a port becomes a profiled-port, all service VFs in that domain are provisioned on this port.

Port-profile backward compatibility

The AMPP port-profile-domain-based provisioning model is different from that in prior releases in two respects:

- The default port-profile configuration is not the same. The **switchport trunk allow vlan all** command that was present in prior releases has been removed. Other related configurations remain the same.
- A user-defined port-profile-domain has been introduced to support VM mobility. A port-profile must be explicitly associated with a profile domain.

In order to maintain legacy AMPP behavior when service VFs are disabled as a result of an upgrade, the following occurs:

- After upgrade, a new port-profile named UpgradedVlanProfile is auto-created. This profile has the single VLAN profile that contains the statement "switch port trunk allow vlan all". This is the configuration that is present in the pre-Network OS 4.1.1 default port-profile to resolve the provisioning differences before the service VFs are enabled.
- After upgrade, a default port-profile domain is created. This default domain contains all the existing user-created port-profiles and vCenter-created auto-profiles prior to the upgrade, in addition to the UpgradedVlanProfile.

The following rules apply to the UpgradedVlanProfile and the default port-profile domain while the switch is in the VF-disabled state after the upgrade.

- The user cannot edit the UpgradedVlanProfile.
- The newly created user port-profile or vCenter auto-profile is automatically added to the default port-profile domain.
- The deleted user or auto port-profile is automatically deleted from the default port-profile domain.
- The **show running-config** command or the **show port-profile domain** command shows the port-profiles in the default-profile-domain.
- The user is not allowed to edit the default port-profile domain.

The following rules apply after service VFs are enabled in the fabric.

- The user can edit the UpgradedVlanProfile just like any other port-profile.
- The newly created port-profile is not automatically added to the default domain. It can only be explicitly added to or removed from the default profile-domain.
- vCenter-managed auto-profiles continue to be added to or deleted automatically from the default port-profile domain.

  NOTE
  In Network OS 4.1.1 and later, the vCenter auto-profile does not support service VF classification.

- The **show running-config** command or the **show port-profile domain** command shows the port-profiles in the port-profile domain.
- The user is allowed to edit the default port-profile domain.
- The user is not allowed to delete the default port-profile-domain.

The following table compares the results of a **show running-config** command before and after an upgrade from Network OS 4.0.0 to Network OS 4.1.1 and later.

TABLE 10 Configuration status before and after upgrade

Network OS 4.0.0	Network OS 4.1.1 and later
port-profile default allow non-profiled-macs vlan-profile switchport switchport mode trunk switchport trunk allowed vlan all switchport trunk native-vlan 1 ! fcoe-profile fcoeport default	port-profile default allow non-profiled-macs vlan-profile switchport switchport mode trunk switchport trunk native-vlan 1 ! fcoe-profile fcoeport default !

TABLE 10 Configuration status before and after upgrade (continued)

Network OS 4.0.0	Network OS 4.1.1 and later
! restrict-flooding ! port-profile pp1 vlan-profile switchport switchport mode trunk switchport trunk allowed vlan add 11 ! ! port-profile pp2 vlan-profile switchport switchport mode trunk switchport trunk allowed vlan add 12 ! !	restrict-flooding ! ! port-profile pp1 vlan-profile switchport switchport mode trunk switchport trunk allowed vlan add 11 ! ! port-profile pp2 vlan-profile switchport switchport mode trunk ! ! port-profile UpgradedVlanProfile vlan-profile switchport switchport mode trunk switchport trunk allowed vlan all ! port-profile-domain default port-profile pp1 port-profile pp2 port-profile UpgradedVlanProfile !

Association of port-profile-domains with an interface

The **port-profile-port** command allows a user to associate a port-profile-domain (default or nondefault) with an interface. The result is that all VLANs specified therein are configured onto the interface.

When the **domain** keyword is not used, the default port-profile-domain is associated with an interface, as in the following example.

```
switch(config)# int te 2/0/1
switch(config-int-te-2/0/1)# port-profile-port
```

STP with service VFs

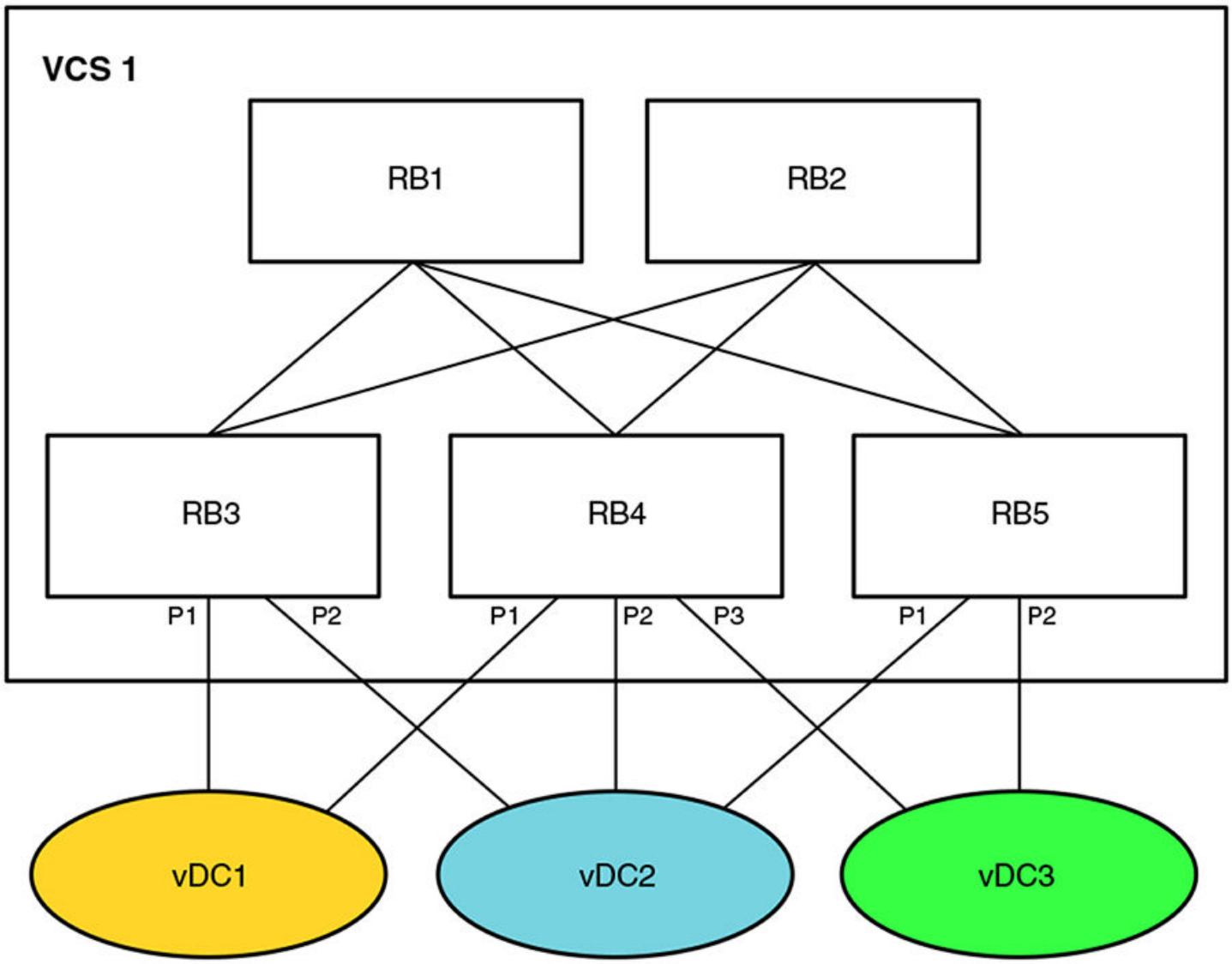
Spanning Tree Protocol (xSTP) is configured on a switch port as in previous releases. However, the user is responsible for configuring xSTP correctly. The user must ensure that VLAN configurations and VLAN-to-instance mappings are consistent across switch ports, and must understand whether RSTP or MSTP are applicable when 802.1Q VLANs and overlapping service VF classifications coexist in the fabric.

A VCS Fabric supports only a single STP domain of any protocol flavor that sends untagged BPDUs (for example, RSTP/MSTP). This domain should contain only switch ports that have identical VLAN configurations, whether 802.1Q or service VF. This is necessary for STP to operate correctly across the fabric. All other switch ports that do not participate in this domain must have STP disabled.

STP-with-service-VFs topology

The following illustrates an example VCS Fabric that is connected to three vDCs. There must not be any external physical connectivity among the vDCs.

FIGURE 25 STP-with-service-VFs topology



STP-with-service-VFs configuration rules

It is the user's responsibility to determine which vDC should participate in RSTP. The following behavior and rules apply:

- RSTP
  - RSTP is VLAN-unaware. Service VF configurations are allowed in each vDC.
  - A single RSTP topology is formed by the VCS Fabric (which appears as a single logical switch) and all attached vDCs.
  - A topology change in one vDC affects the topology of other vDCs.
  - Data loops between the VCS Fabric and individual vDCs are detected by this RSTP topology.
- MSTP
  - The VCS Fabric and the attached vDCs belong to the same MSTP region.
  - VLAN-to-instance mapping must be the same in the VCS Fabric and for each vDC.
  - An MSTP instance topology is formed by the VCS Fabric (which appears as a single logical switch) and all attached vDCs.

- Service VF configuration is allowed. Service VFs (VLAN IDs greater than 4095) are not assigned to any instance and are always in a forwarding state.
- A topology change in one MSTP instance for a vDC affects the topology of the same instance in other vDCs.
- Data loops between the VCS Fabric and individual vDCs are detected by each MSTP topology.

STP participation

The default state for all VLANs (802.1Q or service VFs) is "no spanning-tree shutdown."

For RSTP, there is one RSTP instance in the VCS fabric; the protocol is still VLAN-unaware. This RSTP instance consists of switch ports that have identical VLAN configurations. The VLAN configuration may consist of 802.1Q VLANs or classified VLANs. The STP port state applies to all VLANs (802.1Q and classified) on a port. For switch ports that cannot participate in the same RSTP instance, it is the user's responsibility to shut down spanning tree on these ports. This is the case where ports have overlapping C-TAG classifications and these C-TAGs represent different service VFs.

For MSTP, the VCS Fabric is a single MSTP region. The VLAN-to-instance mapping is applicable only to 802.1Q VLANs. The MSTP VLAN digest calculation is based on 802.1Q VLANs alone. The MSTP state is applied on a port-instance basis (as in previous releases). For switch ports that cannot participate in the same MSTP instance, it is the user's responsibility to shut down spanning tree on these ports. This is the case where ports have overlapping C-TAG classifications.

Service VFs can participate only in MST instance 0 and cannot be assigned to another MST instance. When a service VF is shut down, it is assigned to an internal instance (instance 255) that is always in the forwarding state. The default state for all 802.1Q VLAN and service VFs is "no spanning-tree shutdown."

For PVST, the STP instance is on a per-service-VF basis. PVST can be enabled only on a service VF that has a classification tag. The classification tag identifies the default 802.1Q VLAN in the attached network and is carried in the PVST BPDU; it is also used to form the root RBridge ID. The service VF must have the same C-TAG classification and a nonconflicting classification with other VFs on all RBridge interfaces. This is to ensure the uniqueness of the root RBridge ID for each PVST instance. Consequently, note the following conditions:

- PVST cannot be enabled on a service VF with a VLAN ID greater than 4095 on an access port.
- PVST cannot be enabled on a trunk-mode native VLAN that has no C-TAG classification.

For edge-loop detection (ELD), the protocol instance is on a per-service-VF basis. The user can use the CLI to enable ELD for any service VF on a switch port. Because an ELD configuration applies on a port, the classification for that service VF must exist before the ELD configuration can be accepted.

STP tunneling

BPDU tunneling can be controlled on a per-port basis by means of the existing **bpdu-drop** command. In a multitenancy environment, where a VCS Fabric could be connected to multiple STP-enabled networks, the fabric should tunnel only one instance of the STP BPDU. It is the user's responsibility to enable tunneling on ports that belong to the same STP instance. Other switch ports should have tunneling disabled.

Currently, the **bpdu drop** command controls BPDU forwarding only on the ingress port; the forwarding decision must be applied on the egress port as well. Nontagged BPDUs are tunneled on VCS control VLAN 4095. At the egress RBridge, switch ports that have BPDU tunnels disabled should be removed from the flood membership of the VLAN. For tagged BPDUs (as in PVST), a BPDU is tunneled on its own service-VF flood domain.

PVLANs with service VFs

Private VLAN (PVLAN) configurations apply to service VFs. A service VF can be a primary or a secondary VLAN. However, before an association between the primary and secondary VLAN can be made at the trunk port, the classification of a PVLAN that is a service VF

must have been configured. Based on the PVLAN type, the classification is done at the respective promiscuous or host port. That is, the classification of a primary VLAN is done at a promiscuous port, that of a community VLAN at a community port, and that of an isolated VLAN at an isolated port.

The service VF classification is required only if a port operates in trunk mode. An access port does not require classification rules. The validation that is done for the service VF corresponds to the port type.

IP over service VFs

Layer 3 configurations are applicable to service VFs as well. The **interface ve** command for virtual Ethernet (VE) interfaces also applies to service VFs, and all commands under the **interface ve** submode are supported.

Each VE interface is mapped to a service VF, and all such interfaces share the router's MAC address.

A VE interface can be assigned to a VRF instance. Layer 3 runs per-VRF OSPF instances to exchange routes between RBridges in a given VRF instance.

Transport VFs

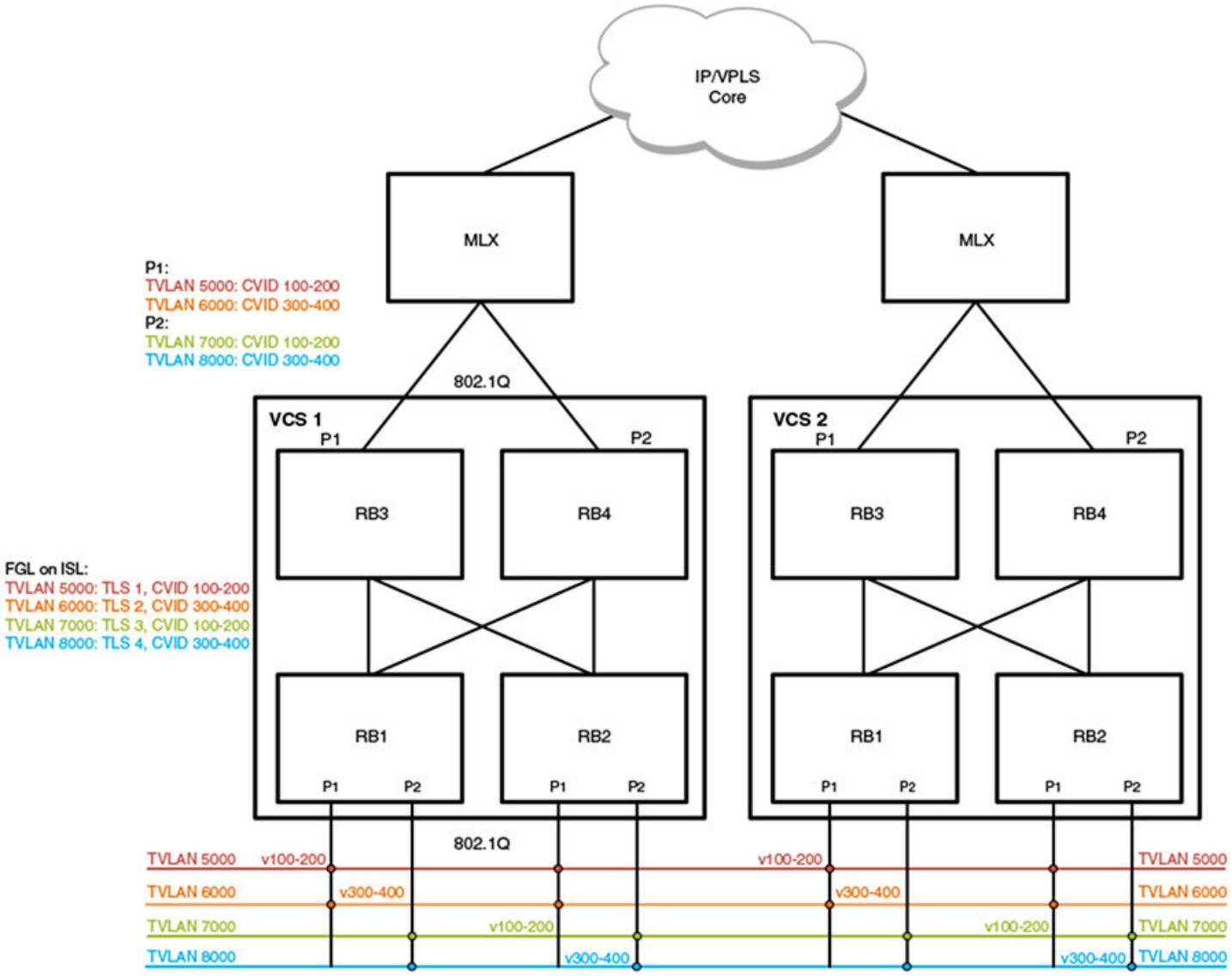
Transport service enables a cloud provider to offer service applicability on a VLAN group level, rather than at a per-individual tenant VLAN level.

The cloud-based service provider needs to provide different service-level agreements (SLAs) to support tenant needs and changes. Transport VFs enable the provider to offer services applicability on a VLAN group level, rather than at a per-individual tenant VLAN level, where the VLAN group represents a specific tenant application. Accordingly, the service offered can be associated only with a single transparent VLAN that collectively represents all the VLANs in the group that participate in supporting a given application. With transport VFs, all VLANs that support the application share the same Layer 2 forwarding domain (individual VLAN isolation is not maintained), and all end stations that participate in a given application or transport VF instance must use unique MAC addresses.

Transport VF topology

The following figure shows four transport VF instances on the respective transparent VLANs (TVLANs) 5000, 6000, 7000 and 8000. Each transport VF carries 101 VLANs in that application. Across the transport VFs, overlapping service VFs among tenants can be supported. The transport VFs can also be extended to another VCS Fabric over a VPLS network.

FIGURE 26 Transport service



The transport VFs that can extend outside of the VCS Fabric are numbered up through 4095, bound by the 802.1Q interface. Because the extension port cannot support QinQ encapsulation, transport VFs that have overlapping C-TAGs cannot be configured on the same port. In the initial release of this feature, no Layer 2 or Layer 3 configuration is supported.

Transport VFs are created by configuring a transparent VLAN that is classified by a set of VLANs at a trunk interface. The operational model is similar to the implementation of SVL (shared VLAN learning). The forwarding behavior is that all service VFs in the transport VF instance are mapped to the transparent LAN forwarding domain (individual VLAN isolation is not maintained), and all end stations participating in the transport VF must have unique MAC addresses.

Transport VF classification rules

The following Transport Layer Security classification rules apply:

- Transport VFs can be configured only on a trunk port, by means of the **transport-service** *tlsid* command in VLAN configuration mode.
- The maximum number of transport VFs that can be configured in this release is 1000.
- Both service and transport VFs can coexist on the same port. A C-TAG is assigned exclusively to a service or transport VF.
- Multiple transport VFs can be configured on the same port.
- Control traffic is classified and handled as follows:
  - Untagged control traffic is not subject to transport VF classification rules. It is handled according to the respective protocol configuration (that is, trapped, dropped, forwarded).
  - Tagged control traffic received on a transport VF is forwarded on the transport VF domain as is data traffic. (PVST cannot be established on a transport VF and is always in the shutdown state.)
  - Tagged control traffic received on a service VF is governed by its respective protocol configuration.
- Transport VF classification can be based on any of the following:
  - A C-TAG range
  - The native VLAN
  - Default traffic (any nonmatching data traffic)
- A vXLAN VNI cannot be mapped to a transport VF.
- Layer 2/Layer 3 configurations are not supported on a transport VF. This means that AMPP/xSTP/PVLAN/RSPAN/ACL/VE configurations are not allowed on a transparent VLAN.

Additional transport VF classification issues

The following table summarizes the classification rules that are applicable to some special VLANs and native VLANs.

TABLE 11 Additional transport VF classification rules

C-TAG	VF classification	Transport VF C-TAG range	Transport VF default VLAN	Notes
Default VLAN 1	No	No	Yes	VLAN 1 cannot be used as a VF classification C-TAG in regular trunk mode. However, it can be used as a VF classification C-TAG in no-default-native-VLAN trunk mode.
Native VLAN	Yes	Yes	Yes	The transport VF default VLAN excludes matching any existing VLAN classification on the port. Because a native VLAN exists as the implicit classification on a trunk port, it is not classified into the default transport VF.
FCoE VLAN (1002)	Yes	Yes	Yes	An FCoE VLAN defined in the FCoE fabric-map configuration can be used as a classification C-TAG if the port is not configured as an FCoE port.
Reserved VLAN	Yes	Yes	Yes	Each platform has a certain VLAN range that is reserved for internal operations. A VLAN in this range can be used as a service or transport C-TAG if the VLAN ID is not internally configured on an edge port. (VLAN 4095 is an internal VLAN and cannot be used.)

Service and transport VF classification with native VLANs

This section addresses two ways to classify service and transport VFs with native VLANs: a default native VLAN mode, and a nondefault native VLAN mode.

Default-native-VLAN trunk mode

When a port is configured in normal trunk mode, a default native VLAN exists. Consequently, the native VLAN complies with the existing native VLAN configuration and forwarding behavior in this mode. The normal behavior for the existing native VLAN is as follows:

- Default native VLAN 1 exists when the port first enters this mode.
- The default native VLAN always exists (either as VLAN 1 or any 802.1Q VLAN ID) and cannot be deleted.
- A tagged native VLAN is always forwarded and cannot be discarded, unless it is blocked by STP.
- Egress tagging behavior depends on acceptable ingress frame types:
  - Tagged egress is enabled only if the acceptable ingress frame type is tagged only.
  - Untagged egress is enabled only if the acceptable ingress frame type includes untagged frames.
  - Egress tagging cannot preserve ingress frame encapsulation.

The following commands are applicable to native VLANs (802.1Q VLANs or service VFs):

- [no] **switchport trunk native vlan** *vid* [**ctag** *ctag*]
- [no] **dot1q tag native** (global configuration mode)
- [no] **switchport trunk tag native-vlan** (interface subtype configuration mode)

The first command is used to define a native 802.1Q VLAN or a native service or transport VF. The last two commands are used to control the ingress acceptable frame and egress tagged behavior. The 802.1Q native VLAN classifications and the role of the respective commands are summarized in the following table.

TABLE 12 802.1Q native VLAN classifications

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI commands
None	N/A	N/A	None (tagged 802.1Q native VLAN is always forwarded)
Untagged only	No	N/A	None (tagged 802.1Q native VLAN is always forwarded)
Tagged only	N/A	Yes	**switchport trunk native vlan** *vid*, **switchport trunk tag native-vlan**, AND **dot1q tag native**
Untagged and tagged	Yes	No	**switchport trunk native vlan** *vid*, **no switchport trunk tag native-vlan**, OR **no dot1q tag native**

The service VF classification rules are similar to those for native VLAN classification, but with the following exceptions:

- VLAN 1 cannot be used as a classification CTAG.
- Ingress and egress tagging behavior is controlled by the interface-level configuration, not by the global configuration.

The following summarizes the native service VF (VLAN ID > 4095) classifications that can or cannot be supported with the respective commands.

TABLE 13 Service VF native VLAN classifications

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI
None	N/A	N/A	None (tagged native VLAN is always forwarded)
Untagged only	No	N/A	**switchport trunk native vlan** *vid*, **no switchport trunk tag native-vlan**
Tagged only	N/A	Yes	**switchport trunk native vlan** *vid* **ctag** *cvid*, **switchport trunk tag native-vlan**
Untagged and tagged	Yes	No	**switchport trunk native vlan** *vid* **ctag** *cvid*, **no switchport trunk tag native-vlan**

The following illustrates configurations that are valid or invalid in regular trunk mode.

```
(config)# int te 5/0/1
(config-if-te-5/0/1)# switchport mode trunk
(config-if-te-5/0/1)# switchport trunk native-vlan-untagged 6000
ERROR: invalid command
# VLAN 1 cannot be used as a classification tag
(config-if-te-5/01)# switchport trunk native-vlan 6000 ctag 1
ERROR: default vlan 1 cannot be classified to a virtual fabric
# CTAG not used elsewhere can be used
(config-if-te-5/01)# switchport trunk native-vlan 6000 ctag 2
# Service VF classification without C-TAG is permitted if ingress frame type allows untagged packet
# at that interface. Global mode tagging control does not apply to a service VF.
(config-if-te-5/01)# switchport trunk native-vlan 7000
ERROR: interface is not configured to accept untagged packet
(config-if-te-5/01)# no switchport trunk tag native-vlan
(config-if-te-5/01)# switchport trunk native-vlan 7000
```

No-default-native-VLAN trunk mode

Another set of native VLAN classifications for service and transport VFs is available in trunk mode without the implicit creation of a default VLAN. The purpose of this trunk mode is to provide flexibility that is not available in default VLAN trunk mode and do the following:

- Provide a raw Layer 2 switchport with no VLAN configuration.
- Allow native VLAN configuration when it is needed.
- Allow the mapping of a native VLAN to a service or transport VF.
- Allow the independent specification of ingress acceptable frame type and egress tagging options.

This trunk mode differs from the default-VLAN trunk mode in the following ways:

- Default VLAN 1 is not implicitly created in this mode.
- Native VLAN commands that are applicable in default-VLAN trunk mode are not supported in this mode.
- Native VLAN commands that are applicable in this mode are not supported in default-VLAN trunk mode.

Because of the different mode behaviors, the user must be aware of the following:

- A port in default-VLAN trunk mode cannot use the new classifications. For example, an AMPP profile port operates in default-VLAN trunk mode and therefore the new classifications are not supported in this case.
- When a port is configured in this mode there is no assumption that VLAN 1 exists. The default VLAN 1 must be configured explicitly. For example, PVST VLAN 1 is not enabled on an interface unless that VLAN is configured explicitly.

The following commands are used to support different native VLAN configurations:

- **switchport mode trunk-no-default-native**

- [no] **switchport trunk native-vlan-untagged** *vid*
- [no] **switchport native-vlan-xtagged** *vid* [**ctag** *cvid*] **egress** [**tagged** | **untagged** | **any**]

The service and transport VF native VLAN classifications and the role of the respective commands are summarized in the following table.

TABLE 14 Service and transport VF native VLAN classifications

Ingress allowed frame type	Egress tagging: Untagged only	Egress tagging: Tagged only	CLI commands
None	N/A	N/A	**switchport mode trunk-no-default-native**
Untagged only	Yes	N/A	**switchport trunk native-vlan-untagged** *vid*
Tagged only	N/A	Yes	Use regular VLAN classification: **switchport trunk allow vlan add** *vid* [**ctag** *cvid*]
Untagged and tagged	Yes	Yes	**switchport trunk native-vlan-xtagged** *vid* [**ctag** *cvid*] [**ctag** *cvid*] **egress** [**untagged** | **tagged** | **any**]

The following configuration rules apply in no-default-native-VLAN trunk mode.

Rules for the **switchport trunk-no-default-native** command:

- There is no automatic native VLAN configuration.
  - VLAN 1 is not created automatically when the port is configured in this mode.
  - VLAN 1 is not a default VLAN. It is just another C-TAG that is available for classification.
  - The user can explicitly create VLAN 1 to achieve a default-VLAN trunk mode configuration, by using the **switchport trunk allow vlan add** *vid* command.
- An untagged or unclassified frame is discarded by default.
- All switchport configurations except the following native VLAN commands in default-VLAN trunk mode continue to be supported:
  - **switchport trunk tag native-vlan**
  - **switchport trunk native vlan** *vlan_id*
  - **dot1q tag native-vlan** (a global command that does not apply to a port)
- All service and transport VF configurations that are available in default-VLAN trunk mode continue to be supported. An 802.1Q native VLAN can be classified to a service or transport VF with the new commands.

Rules for the [**no**] **switchport native-vlan-untagged** *vid* command are as follows:

- This command accepts untagged packets only and allows egress untagged packets. The VLAN ID can be a regular 802.1Q VLAN ID or a service or transport VF.
- The **no** form of this command removes the native VLAN classification, and untagged frames are dropped.

Rules for the [**no**] **switchport native-vlan-xtgagged** *vid* [**ctag** *cvid*] **egress** [**tagged** | **untagged** | **any**] command are as follows:

- This command accepts untagged or tagged 802.1Q VLANs and service and transport VFs and specifies egress tagging behavior. If a VLAN is an 802.1Q VLAN or a service VF, the supported egress tagging behavior is untagged or tagged. If a VLAN is a transport VF, the **egress** option must be **any**, to indicate that the ingress frame encapsulation must be preserved.
- The **no** form of this command removes the native VLAN classification. Untagged frames and the associated tagged 802.1Q VLAN are dropped.

The following illustrates configuration in no-default-native-VLAN trunk mode.

```
switch(config)# int vlan 5000
switch(config)# int vlan 6000
switch(config-Vlan-6000)# transport-service 60
switch(config)# int vlan 7000
switch(config-Vlan-7000)# transport-service 70
```

In the new mode, the default behavior is to drop all packets.

```
(config)# int te 1/0/1
(config-if-te-1/0/1)# switchport mode trunk-no-default-native
```

VLAN configuration similar to that of regular trunk mode can be achieved explicitly after VLAN 1 is configured as a tagged VLAN.

```
switch(config-if-te-1/0/1)# switchport trunk allow vlan add 1
```

The following creates native VLAN 10. All service and transport VFs can continue to coexist on the same port.

```
switch(config-if-te-1/0/1)# switchport trunk native-vlan-untagged 10
switch(config-if-te-1/0/1)# switchport trunk allow vlan 6000 ctag 100-200
switch(config-if-te-1/0/1)# switchport trunk default-vlan 7000
```

The following accepts ingress tagged or untagged frames, but egress frames must be tagged only. This is not allowed in default-VLAN trunk mode.

```
switch(config-if-te-1/0/1)# switchport trunk native-vlan-xtagged 10 egress tagged
```

The following classifies a tagged native VLAN to service VF 5000.

```
switch(config-if-te-1/0/1)# switchport trunk native-vlan-xtagged 5000 ctag 10 egress tagged
```

The following classifies VLAN 10 to transport VF 6000. Because this native VLAN is a transport VF, the option for **egress** is **any.**

```
switch(config-if-te-1/0/1)# switchport trunk native-vlan-xtagged 6000 ctag 10 egress any
switch(config-if-te-1/0/1)# switchport trunk allow vlan 6000 ctag 100-200
```

The following configurations are valid or invalid in no-default-native-VLAN mode.

```
switch(config)# int te 5/0/1
switch(config-if-te-5/0/1)# switchport mode trunk-no-default-native
switch(config-it-te-5/0/1)# switchport trunk tag-native
    ERROR: Port mode not set to trunk
switch(config-it-te-5/0/1)# switchport trunk native vlan 2
    ERROR: Port mode not set to trunk
    # C-TAG classification to service or transport VF mapping is still 1 to 1.
    # Rejected as duplicate classification even when same CTAG-to-VF mapping is given.
switch(config-if-te-5/0/1)# switchport trunk native-vlan-xtagged 6000 ctag 10 egress any
switch(config-if-te-5/0/1)# switchport trunk allow vlan add 6000 ctag 10-20
    ERROR: ctag already used in other classification
switch(config-if-te-5/0/1)# switchport trunk allow vlan 7000 ctag 10
    ERROR: ctag already used in other classification
```

Configuring and managing Virtual Fabrics

The fabric must be in logical chassis cluster mode for the configuration and management of the Virtual Fabrics feature. Whenever changes are made, they are saved automatically. Ensure that the fabric is in logical chassis cluster mode before attempting to configure or manage this feature.

Refer also to Virtual Fabrics overview on page 103.

Configuring a service VF instance

Configuring a service VF instance consists of enabling VF configuration in the fabric, and then configuring a service VF instance that is greater than 4095, The initial release of this feature supports up through 8192 VLANs, with 8191 being the largest number that can be assigned.

The **vcs virtual-fabric enable** command , issued in global configuration mode, expands the VLAN ID address space beyond the 802.1Q limit in the fabric, allowing VLANs with IDs greater than 4095 to be supported, up through 8191. The command, which is accepted only if the state of the fabric is capable of supporting VLAN virtualization, does not disrupt existing 802.1Q data traffic in the fabric.

Enabling service VF configuration

In global configuration mode, issue the **vcs virtual-fabric enable** command:

```
switch(config)# vcs virtual-fabric enable
```

NOTE

Use the **no** form of this command to disable service VF configuration.

Creating a service VF instance

In global configuration mode, use the **interface vlan** *vlan_id* command, where *vlan_id* is a number greater than 4095, through 8191, and is not a reserved VLAN.

```
switch(config)# interface vlan 5000
```

NOTE

Use **no interface vlan** *vlan_id* to delete a service VF.

Configuring a transport VF instance

The following example command sequence illustrates the configuration of three transport VF instances.

```
switch(config)# interface vlan 6001
switch(config-Vlan-6001)# transport-service 1
switch(config)# interface vlan 6002
switch(config-Vlan-6002)# transport-service 2
switch(config)# interface vlan 6003
switch(config-Vlan-6003)# transport-service 3
```

Configuring VF classification to a trunk interface

The following example command sequence illustrates the configuration of VF classification to a trunk interface.

```
switch(config)# interface TenGigabitEthernet 1/0/1
switch(conf-if-te-1/0/1)# switchport
switch(conf-if-te-1/0/1)# switchport mode trunk
switch(conf-if-te-1/0/1)# switchport trunk allowed vlan add 5000 ctag 100
switch(conf-if-te-1/0/1)# switchport trunk allowed vlan add 5001 ctag 101
```

Configuring transport VF classification to a trunk interface

The following example command sequence illustrates the configuration of VF classification to a trunk interface.

```
switch(config)# interface TenGigabitEthernet 1/0/1
switch(conf-if-te-1/0/1)# switchport
switch(conf-if-te-1/0/1)# switchport mode trunk
switch(conf-if-te-1/0/1)# switchport trunk allowed vlan add 6000 ctag 600-610
```

Creating a default VLAN with a transport VF to a trunk interface

The following example command sequence illustrates the configuration of a default VLAN with a transport VF to a trunk interface.

```
switch(config)# interface TenGigabitEthernet 1/0/1
switch(conf-if-te-1/0/1)# switchport
switch(conf-if-te-1/0/1)# switchport mode trunk
switch(conf-if-te-1/0/1)# switchport trunk default-vlan 7000
```

Configuring a native VLAN in regular VLAN trunk mode

The following examples illustrate the configuration of a native VLAN in regular VLAN trunk mode, where the default native VLAN 1 exists.

The following native VLAN commands are supported in this trunk mode:

- **switchport trunk tag native-vlan**
- **switchport trunk native-vlan**

1. Create native VLAN 10.

   NOTE
   VLAN 1 is the default VLAN in this mode.

```
switch(config)# interface te 2/1/1
switch(config-if-te-2/1/1)# switchport mode trunk
switch(config-if-te-2/1/1)# switchport trunk native-vlan 10
```

2. Configure untagged native VLAN (service VF) 5000 and transport VLAN (transport VF) 6000, and make VLAN 7000 the default VLAN.

```
switch(config)# interface te 2/1/1
switch(config-if-te-2/1/1)# switchport mode trunk
switch(config-if-te-2/1/1)# no switchport trunk tag native
switch(config-if-te-2/1/1)# switchport trunk native-vlan 5000
switch(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
switch(config-if-te-2/1/1)# switchport trunk default-vlan 7000
```

3. Configure transport VLAN 6000 to accept the C-TAG range 100 through 200, as well as tagged or untagged native VLAN traffic.

```
switch(config)# interface te 2/1/1
switch(config-if-te-2/1/1)# switchport mode trunk
switch(config-if-te-2/1/1)# no switchport trunk tag native
switch(config-if-te-2/1/1)# switchport trunk native-vlan 6000 ctag 10
switch(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
```

Configuring a native VLAN in no-default-native-VLAN trunk mode

The following examples illustrate the configuration of a native VLAN in a trunk mode where the native VLAN does not exist. The native VLAN must be explicitly created in this mode.

The following native VLAN commands are supported in this mode:

- **switchport trunk native-vlan-untagged**
- **switchport trunk native-vlan-xtagged**

1. Configure a trunk port without a default native VLAN, then explicitly configure the native VLAN.

```
switch(config)# interface te 2/1/1
switch(config-if-te-2/1/1)# switchport mode trunk-no-default-native
switch(config-if-te-2/1/1)# switchport trunk native-vlan-xtagged 10 egress tagged
```

2. In no-default-native-VLAN trunk mode, configure untagged native VLAN 5000 and transport VLAN 6000, and make VLAN 7000 the default VLAN.

```
switch(config)# interface te 2/1/1
switch(config-if-te-2/1/1)# switchport mode trunk-no-default-native
switch(config-if-te-2/1/1)# switchport trunk native-vlan-untagged 5000
switch(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
switch(config-if-te-2/1/1)# switchport trunk default-vlan 7000
```

3. In no-default-native-VLAN trunk mode, configure transport VLAN 6000 to accept C-TAG range 100 through 200 as well as tagged or untagged native VLAN traffic..

```
switch(config)# interface te 2/1/1
switch(config-if-te-2/1/1)# switchport mode trunk-no-default-native
switch(config-if-te-2/1/1)# switchport trunk native-vlan-xtagged 6000 ctag 10 egress any
switch(config-if-te-2/1/1)# switchport trunk allow vlan add 6000 ctag 100-200
```

Configuring additional Layer 2 service VF features

This section addresses additional features that are available on trunk ports once a Virtual Fabric is established.

Configuring private service VFs

The following examples illustrate the configuration of a variety of private VLANs (PVLANs) as service VFs and their association on access and trunk ports.

Refer also to PVLANs with service VFs on page 121.

Configuring service VFs and defining and associating PVLANs

1. In global configuration mode, use the **interface vlan** *vlan_id* command to create VLAN instances that are greater than 4095, through 8191.

```
switch(config)# interface vlan 5000
switch(config)# interface vlan 6000
switch(config)# interface vlan 7000
```

2. In VLAN configuration mode, use the **private-vlan** command to create the three types of PVLAN.

```
switch(config)# interface vlan 5000
switch(conf-vlan-5000)# private-vlan primary
switch(config)# interface vlan 6000
switch(conf-vlan-6000)# private-vlan isolated
switch(config)# interface vlan 7000
switch(conf-vlan-7000)# private-vlan community
```

3. Using the **private-vlan association** command, associate the secondary PVLANs with the primary PVLAN.

```
switch(config)# interface vlan 5000
switch(conf-vlan-5000)# private-vlan association add 6000
switch(conf-vlan-5000)# private-vlan association add 7000
```

Configuring physical interfaces

1. Create classification rules for the primary and secondary VLAN at the respective primary and host ports.

   The classification must be done before the primary-to-secondary VLAN associations are specified. In following example, the same C-TAG is used to classify the primary and secondary VLANs.

   Interface te 1/0/1 is a primary trunk port.

   ```
   switch(config)# interface te 1/0/1
   switch(conf-if-te-1/0/1)# switchport mode private-vlan trunk promiscuous
   switch(conf-if-te-1/0/1)# switchport trunk allow vlan add 5000 ctag 10
   ```

   Interface te 1/0/2 is an isolated trunk port.

   ```
   switch(config)# interface te 1/0/2
   switch(conf-if-te-1/0/2)# switchport mode private-vlan trunk host
   switch(conf-if-te-1/0/2)# switchport trunk allow vlan add 6000 ctag 10
   ```

   Interface te 0/3 is a community trunk port.

   ```
   switch(config)# interface te 1/0/3
   switch(conf-if-te-1/0/3)# switchport mode private-vlan trunk host
   switch(conf-if-te-1/0/3)# switchport trunk allow vlan add 7000 ctag 10
   ```

2. Configure the PVLAN association on the promiscuous trunk port.

   ```
   switch(config)# interface te 1/0/1
   switch(conf-if-te-1/0/1)# switchport mode private-vlan trunk promiscuous
   switch(conf-if-te-1/0/1)# switchport trunk allowed vlan add 400
   switch(conf-if-te-1/0/1)# switchport trunk allowed vlan add 5000 ctag 10
   switch(conf-if-te-1/0/1)# switchport private-vlan mapping 5000 add 6000
   switch(conf-if-te-1/0/1)# switchport private-vlan mapping 5000 add 7000
   ```

3. Configure the PVLAN association on the promiscuous access port.

   ```
   switch(config)# interface te 1/1/1
   switch(conf-if-te-1/1/1)# switchport mode private-vlan promiscuous
   switch(conf-if-te-1/1/1)# switchport private-vlan mapping 5000 add 6000
   switch(conf-if-te-1/1/1)# switchport private-vlan mapping 5000 add 7000
   ```

4. Configure the isolated PVLAN on the trunk port.

   ```
   switch(config)# interface te 1/0/2
   switch(conf-if-te-1/0/2)# switchport mode private-vlan trunk host
   switch(conf-if-te-1/0/2)# switchport private-vlan host-association 5000 6000
   ```

5. Configure the isolated PVLAN on the access port.

   ```
   switch(config)# interface te 1/1/2
   switch(conf-if-te-1/1/2)# switchport mode private-vlan host
   switch(conf-if-te-1/1/2)# switchport private-vlan host-association 5000 6000
   ```

6. Configure the community PVLAN on the trunk port.

   ```
   switch(config)# interface te 1/0/3
   switch(conf-if-te-1/0/3)# switchport mode private-vlan trunk host
   switch(conf-if-te-1/0/3)# switchport trunk allowed vlan add 7000 ctag 10
   switch(conf-if-te-1/0/3)# switchport private-vlan host-association 5000 7000
   ```

7. Configure the community PVLAN on the access port.

   ```
   switch(config)# interface te 1/1/3
   switch(config-if-te-1/1/3)# switchport mode private-vlan host
   switch(config-if-te-1/1/3)# switchport mode private-vlan host 5000 7000
   ```

8. Configure the PVLAN trunk port.

```
switch(config)# interface te 1/4/1
switch(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 5000 ctag 10
switch(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 6000 ctag 20
switch(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 7000 ctag 30
switch(conf-if-te-1/4/1)# switchport mode private-vlan trunk
```

The trunk port can have both PVLAN and regular VF configurations. The following configures PVLAN VFs.

```
switch(conf-if-te-1/4/1)# switchport trunk allowed vlan add 5000 ctag 10
switch(conf-if-te-1/4/1)# switchport trunk allowed vlan add 6000 ctag 10
switch(conf-if-te-1/4/1)# switchport trunk allowed vlan add 6000 ctag 10
switch(conf-if-te-1/4/1)# switchport private-vlan association trunk 5000 6000
switch(conf-if-te-1/4/1)# switchport private-vlan association trunk 5000 7000
```

The following configures non-PVLAN VFs.

```
switch(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 400
switch(conf-if-te-1/4/1)# switchport private-vlan trunk allowed vlan add 5000 ctag 100
```

Understanding PVLAN configuration failures

The following example conditions, with error messages, that determine the success or failure of a PVLAN configuration.

- If a PVLAN is a service VF and the classification does not exist for this port, the PVLAN association executed on this interface will fail. This following commands fail because there is no classification rule for primary service VF 5000.

```
switch(config)# interface te 1/0/1
switch(conf-if-te-1/0/1)# switchport mode private-vlan trunk promiscuous
switch(conf-if-te-1/0/1)# switchport private-vlan mapping 5000 add 200
%%ERROR: Primary Vlan should have a ctag associated with it on this port.
switch(config)# interface te 1/0/2
switch(conf-if-te-1/0/2)# switchport mode private-vlan trunk host
switch(conf-if-te-1/0/2)# switchport private-vlan host-association 100 add 6000
%%ERROR: Secondary Vlan should have a ctag associated with it on this port.
```

- The following command succeeds because the primary port is an access port.

```
switch(conf-if-te-1/1/1)# switchport mode private-vlan promiscuous
switch(conf-if-te-1/1/1)# switchport private-vlan mapping 5000 add 6000
```

- The following command succeeds because the secondary port is an access port.

```
switch(conf-if-te-1/1/2)# switchport mode private-vlan host
switch(conf-if-te-1/1/2)# switchport private-vlan host-association 5000 add 6000
```

Configuring MAC groups

You can assign individual MAC addresses or a group of MAC addresses to a service VF at an access port.

Creating a MAC group instance and assigning MAC addresses

1. In global configuration mode, create a MAC group instance to define the MAC addresses of end stations, by using the **mac-group** *mac-group-id* command.

   The value of *mac-group-id* ranges from 1 through 500.

```
switch(config)# mac-group 1
```

2. In MAC group configuration mode, use the **mac** *mac_address* command to add a MAC address in hexadecimal notation.

NOTE

Ranging is not allowed. Leading zeros can be omitted.

```
switch(config-mac-group 1)# mac 0002.0002.0002
```

Deleting a MAC group or MAC address

1. This command deletes a MAC group.

```
switch(config)# no mac-group 1
```

2. This command deletes a MAC address from a MAC group.

NOTE

Only one MAC address can be deleted at a time.

```
switch(config)# mac-group 1
switch(config-mac-group 1)# no mac 0004.0004.0004
```

Configuring an interface for service VF MAC address access

The following illustrates various options and errors that can occur in configuring an interface for service VF MAC address access.

1. In interface configuration mode, set switchport mode to access and change the default VLAN to a service VF.

NOTE

The default VLAN must be unique. It must not be the same as that used for another MAC classification.

```
switch(config)# int te 2/0/1
switch(config-if-te-2/0/1)# switchport access vlan 5000
```

2. Classify the access VLAN by means of a MAC address.

```
switch(config-if-te-2/0/1)# switchport access vlan 6000 mac 0002.0002.0002
```

3. Classify another access VLAN on the same interface by means of a MAC address.

```
switch(config-if-te-2/0/1)# switchport access vlan 7000 mac 0004.0004.0004
```

NOTE

Frames that do not match 0002.0002.0002 or 0004.0004.0004 are classified into service VF 5000.

4. Create a MAC group to be used to classify a VLAN.

```
switch(config)# mac-group 1
switch(config-mac-group 1)# mac 0002.0002.0002
switch(config-mac-group 1)# mac 0005.0005.0005
switch(config-mac-group 1)# mac 0008.0008.0008
```

5. Configure another service VF on the same interface by means of a MAC group.

```
switch(config-if-te-2/0/1)# switchport access vlan 7000 mac-group 1
Error: mac address is already used in another classification
```

NOTE

The MAC address cannot be used in another service VF classification.

6. Configure another interface with a third service VF and classify the VLANs by MAC group and MAC address, respectively.

```
switch(config-if-te-3/0/1)# switchport mode access
switch(config-if-te-3/0/1)# switchport access vlan 7000 mac-group 1
```

```
switch(config-if-te-3/0/1)# switchport access vlan 8000 0008.0008.0008
switch(config-if-te-3/0/1)# %Error: Mac-address/Mac-group is overlapping with another Mac-address/
Mac-group configuration on the same port.
```

NOTE

The MAC address cannot be used in another service VF classification.

Configuring Layer 3 service VF features

Refer to IP over service VFs on page 122.

Layer 3 configurations are applicable to service VFs, by means of existing **interface ve** commands. Each virtual Ethernet (VE) interface is mapped to a service VF, and all VE interfaces share the router's MAC address. A VE interface can be assigned to a VRF instance, with per-VRF OSPF instances exchanging routes between RBridges in that VRF instance. Virtual Router Redundancy Protocol (VRRP) provides high availability.

Do the following to configure Layer 3 service VF features.

1. Create a service VF and add it to the trunk port with a C-TAG.

```
switch(config)# interface vlan 5000
switch(config)# interface te 3/1/1
switch(conf-if-te-3/1/1)# switchport
switch(conf-if-te-3/1/1)# switchport mode trunk
switch(conf-if-te-3/1/1)# switchport trunk allowed vlan add 5000 ctag 50
```

2. Enable VRF and VRRP on an RBridge.

```
switch(config)# rbridge-id 3
switch(config-rbridge-id-3)# protocol vrrp
```

3. Create a VRF instance and set OSPF routing parameters.

NOTE

BGP is also supported in the appropriate context.

```
switch(config)# ip vrf VRF_CUST1
```

a) Use the **rd** (Route Distinguisher) command to assign an administrative number.

```
switch(config-vrf-CUST1)# rd 10.1.1.1:1
```

b) Set the OSPF address family to IPv4.

```
switch(config-vrf-CUST1)# address-family ipv4
switch(config-vrf-CUST1)# exit
```

c) Return to config mode.

```
switch(config-vrf-CUST1)# exit
```

4. Create the VE interface on the service VF and configure VRF forwarding and VRRP.

a) Configure forwarding for a VRF instance.

```
switch(config-ve-5000)# ip vrf forwarding VRF_CUST1
```

b) In RBridge ID configuration mode, create a virtual Ethernet interface, and assign the service VF and IP address. Be sure to enable the interface.

```
switch(config)# rbridge-id 3
switch(config-rbridge-id-3)# interface ve 5000
```

```
switch(config-ve-5000)# ip address 10.1.1.1/24
switch(config-ve-5000)# no shutdown
```

c) Create a VRRP group and assign a virtual IP address.

```
switch(config-ve-5000)# vrrp-group 22
switch(config-vrrp-group-22)# virtual-ip 10.1.1.1
```

Configuring Layer 2 extension over Layer 3 with Virtual Fabrics

The VLAN-based broadcast domain is extended over a Layer 3 network without the need for an orchestrator.

The extension function commonly resides at the aggregation layer, either embodied in the aggregation node itself, or hanging as a service off the aggregation node in a one-arm topology. This feature resides in the network spine, with up to four RBridges participating. Tunnels are provided through router ports and switch ports.

In addition, the user can collect statistics on tunnel and gateway traffic, through the range of attached VLANs. Both transmit and receive traffic can be monitored for a single tunnel and range of VLANs, or all tunnels and a range of VLANs. Support is also provided for similar sFlow monitoring.

Supported and unsupported features

The following new features are supported:

- Layer 3 forwarding for extended Layer 2 segments, for connections local to the fabric or through a tunnel
- Extension as part of the main fabric or through a "1-arm" topology in which VLAN and VXLAN traffic occupies the same path through connected subnets
- Ability to use VLANs, service VFs, and transport VFs for extension, in a flexible deployment model
- Nondefault VRFs
- Fully meshed tunnels between fabrics, with BUM through headend replication
- Multicast (without multicast traffic optimization)
- Split horizon in the data plane for loop avoidance
- MAC address learning on tunnels
- Layer 2 fault domain isolation across fabrics
- Layer 3 protocol isolation controls (off by default)
- MAC, IPv4, and IPv6 ingress ACLs
- sFlow configurations

The following features are not supported:

- VXLAN-to-VLAN routing, and vice versa
- VNI translation (because of a single VNI domain)
- Exposing of ARPs behind the physical network as a unicast map
- VXLAN transit
- VLAG as undelay
- Loopback IP as VTEP IP
- BUM optimization (including ARP-related optimization)
- Loop-detection protocols over tunnels. There is no support for STP or tunneling STP BPDUs. (The user is responsible for avoiding loops over any interfaces.)

- Keepalives on tunnels
- TRILL with VXLAN
- Tunnels as a SPAN destination
- vLAG of tunnels to the same remote VTEP
- PACL on tunnels (VACL is supported)
- Tunnels as profile-ports. (However, a VLAN that spans a tunnel can have a profiled physical port, as in the previous release.)
- Router port IP addresses or VE IP addresses used as VTEP addresses
- Fabric cluster mode

Note the following considerations with respect to ACLs, SPAN, sFlow, and statistics, as these features share the ternary content-addressable memory (TCAM) tables for the corresponding features of nontunnel traffic:

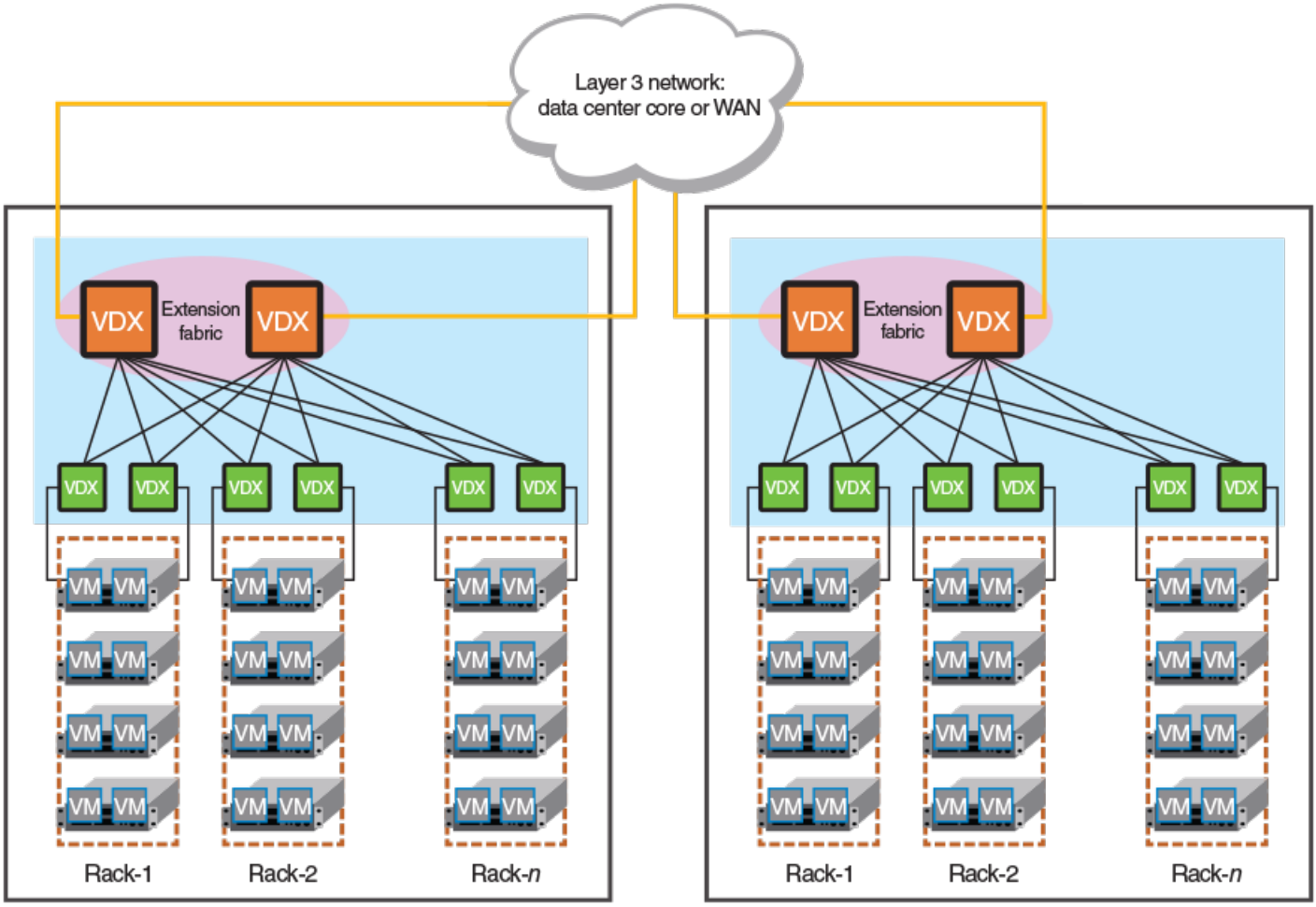
- Ingress statistics and sFLow both need a counting ability. Consequently, if both functions are enabled and a particular traffic type satisfies both classifiers, then statistics alone takes effect and sFlow does not.
- Ingress SPAN and sFlow both use the same resources. Consequently, if both functions are enabled and a particular traffic type satisfies both classifiers, then sFlow takes effect and SPAN does not.
- Ingress statistics and ACLs both use the same resources. Consequently, if both functions are enabled and a particular traffic type satisfies both classifiers, then ACLs take effect and statistics does not.
- The scaling limits of the above features are constrained by the profiles applied on the system.

High-level topologies

The following illustrates an example high-level topology for Layer 2 extension at the aggregation layer only. Note the following restrictions for this topology:

- If a vLAG is deployed, the number of aggregation VDX devices supporting extension is limited to the size of the vLAG with respect to RBridges and the number of RBridges in a VRRP group. (This restriction does not apply if router ports are used.)
- A static route must be configured on the upstream device, or VRRP must use the First Hop Redundancy Protocol (FHRP) gateway.

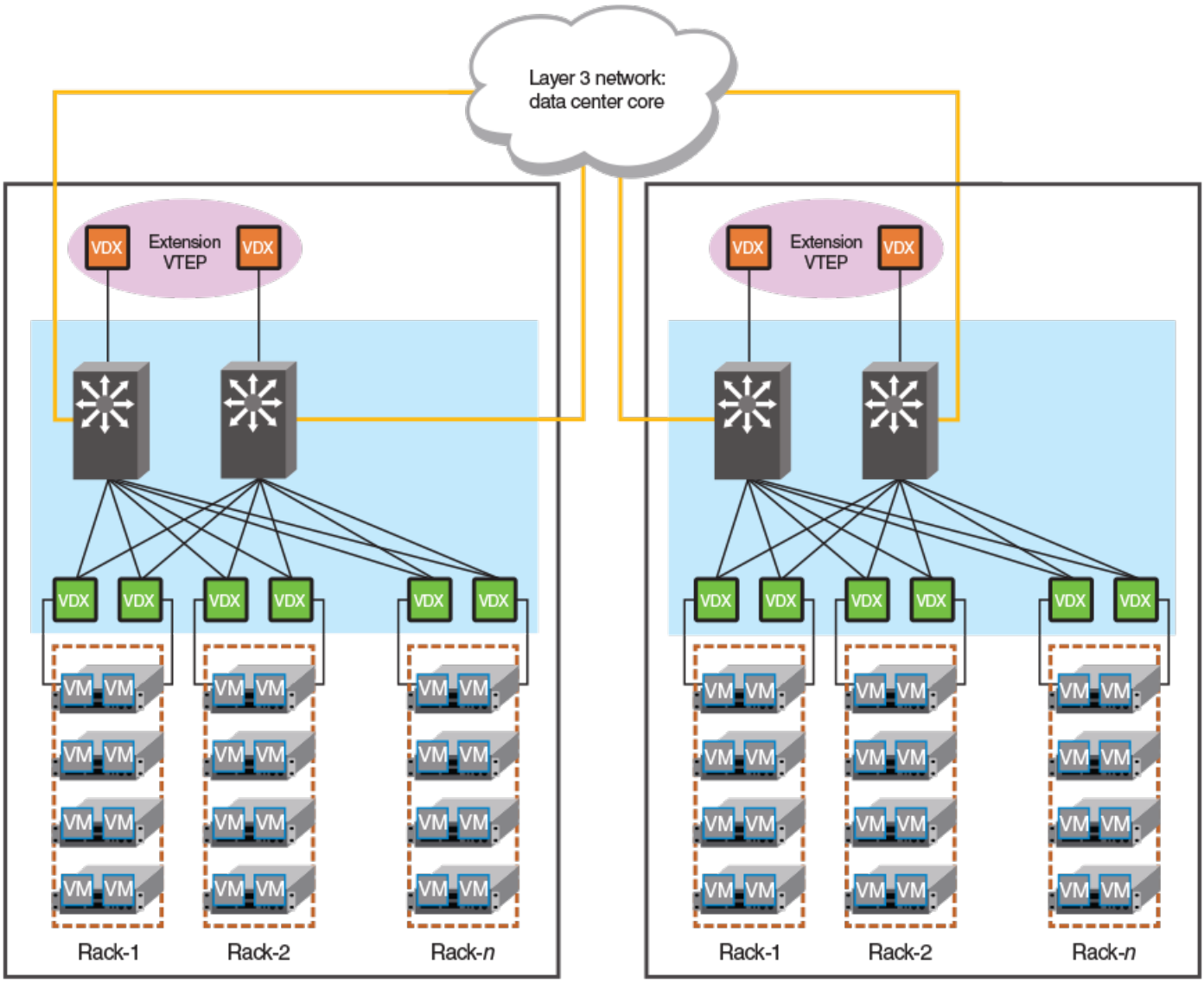
FIGURE 27 Layer 2 extension at the aggregation layer only



The following illustrates an example high-level topology that uses VXLAN virtual network interface (VNI) tunnel endpoints (VTEPs) in separate fabrics with a one-arm VDX deployment. Note the following restrictions for this topology:

- The fabric must be extended through an 802.1Q port.
- Only vLAG connectivity is supported. Router ports are not supported to the aggregation device.

FIGURE 28 Layer 2 extension with VTEPs in separate fabrics (one arm)



VXLAN overlay gateway for Layer 2 extension: summary of steps

This overview summarizes the basic steps that are required to extend a Layer 2 VLAN-based broadcast domain over a Layer 3 network, without the use of an orchestrator such as the NSX Controller.

The basic provisioning and configuration steps can be summarized as follows:

1. Ensure that the network administrator has created a broadcast domain on the physical side. This domain consists of 802.1Q VLANs, as well as the service and transport VFs available in a Virtual Fabrics context.
2. Create a VXLAN tunnel endpoint (VTEP) container with a "type" qualifier that specifies the use of the VTEP for Layer 2 extension.

NOTE
If you are configuring a loopback interface to serve as a VTEP, you must manually configure distinct router IDs, by means of the **ip router-id** command, for use by routing protocols.

3. Create containers for the remote sites. These containers will have the IP addresses of the site. This leads to the automatic creation of VTEP tunnels to the remote sites. The tunnels refer to VTEP as the source endpoint, and to the IP addresses of the remote site as the destination endpoint. (Multiple addresses, corresponding to a LAG of tunnels, are not supported in this release.}

   NOTE
   A full mesh of tunnels must be created.

4. A VLAN-to-VNI map is created for the VTEP. This provides a global mapping across all sites for the VTEP. The map can be created automatically, where VLAN-to-VNI mapping is autogenerated, or the user can specify each map explicitly.
5. In the site container, extend the required VLANs (including service and transport VFs) that must be extended to a given remote site.

Configuring the VXLAN overlay gateway for Layer 2 extension

The network administrator must first configure a broadcast domain.

This task configures the nondefault mode of configuration for a VXLAN overlay gateway.

1. In VXLAN overlay gateway configuration mode, set the type of gateway to support Layer 2 extension.

```
switch(config)# overlay-gateway gateway1
switch(config-overlay-gw-gateway1)# type layer2-extension
```

2. **Using automatic VLAN-to-VNI mapping**: You can optionally use the **map vlan vni auto** command to configure VLAN-to-VNI mappings automatically if they have not been configured manually.

```
switch(config-overlay-gw-gateway1)# map vlan vni auto
```

3. **Using explicit VLAN-to-VNI mapping**: The following configures VLAN-to-VNI mappings manually.

```
switch(config-overlay-gw-gateway1)# map 10,20-22 vni 5000-5002,6000
```

   NOTE
   Automatic and explicit mappings are mutually exclusive. Also, the VLANs used in the **extend vlan** configuration must be a subset of the VLANs used in the **map vlan** configuration. In other words, a VLAN must have a VNI mapping for it to be extended across sites.

4. Configure the site, which includes providing a site name and entering VXLAN overlay gateway site configuration mode, providing an IPv4 address for the VNI tunnel endpoint (VTEP) in that mode, extending VLANs, and enabling the tunnels administratively.
   a) Enter a site name and enter VXLAN overlay gateway site configuration mode.

```
switch(config-overlay-gw-gateway1)# site mysite1
switch(config-overlay-gw-gateway1-mysite1)#
```

   b) Configure an IPv4 address for the tunnel endpoints.

```
switch(config-overlay-gw-gateway1-mysite1)# ip address 10.10.10.1
```

   NOTE
   Only one IPv4 address is allowed.

c) Extend the VLANs.

```
switch(config-overlay-gw-gateway1-mysite1)# extend vlan add 10,20-22
```

d) Enable the tunnels administratively.

```
switch(config-overlay-gw-gateway1-mysite1)# no shutdown
```

5. The following substeps attach the RBridge IDs to the VXLAN gateway instance, set the IP address of the gateway, and activate the instance.

a) Enter the **attach rbridge-id** command to attach existing RBridge IDs to this VXLAN gateway instance. You can specify a range of RBridge IDs up to a maximum of four.

```
switch(config-overlay-gw-gateway1)# attach rbridge-id add 1-2
```

b) Enter the **ip interface ve** *veid* **vrrp-extended-group** *group-ID* command to set the IP address of the VXLAN overlay gateway, as shown in the following example.

```
switch(config-overlay-gw-gateway1)# ip interface ve 10 vrrp-extended-group
100
```

The command accepts the VE interface ID and VRPP-E group ID, then sets the VXLAN overlay gateway's IP address as identical to the already configured VRRP-E virtual IP address. Tunnels that form use this IP address as the source IP address for outgoing packets.

c) Enter the **activate** command to activate this gateway instance.

```
switch(config-overlay-gw-gateway1)# activate
```

This enables all tunnels associated with this gateway. VXLAN tunnels are not user configurable.

d) Return to privileged EXEC mode.

```
switch(config-overlay-gw-gateway1)# end
switch(config)# end
switch#
```

Configuring MAC learning of Layer 2 extension site through BGP

The user can choose the way MAC learning is achieved for a Layer 2 extension tunnel.

BGP routing must be configured.

Data plane (Layer 2) MAC address learning is enabled by default at the remote site. However, this leads to scalability issues. Where BGP routing is used, do the following to enable MAC learning by means of BGP instead. This delegates the responsibility for MAC learning on a tunnel to the Layer 3 control-plane protocol, such as BGP EVPN.

1. Enter VXLAN overlay-gateway site configuration mode.

```
device# config terminal
device(config)# overlay-gateway gateway1
device(config-overlay-gw-gateway1)# site mysite
device(config-overlay-gw-gateway1-site-mysite)#
```

2. Enter the **mac-learning protocol bgp** command.

```
device(config-overlay-gw-gateway1-site-mysite)# mac-learning protocol bgp
```

3. To disable BGP MAC learning and return to the default of Layer 2 learning, use the **no mac-learning protocol bgp** command.

```
device(config-overlay-gw-gateway1-site-mysite)# no mac-learning protocol bgp
```

Additional commands for VXLAN configuration

Additional commands that support VXLAN configuration are listed in the following table.

> NOTE
> For complete information on the those commands as well as other VXLAN overlay-gateway commands and commands related to the NSX Controller, refer to the *Network OS Command Reference*.

TABLE 15 Additional commands for VXLAN configuration

Command	Description
clear overlay-gateway	Clears counters for the specified gateway.
enable statistics	Enables statistics for tunnels.
sflow remote-endpoint	Applies an sFlow profile for a VXLAN overlay gateway and sets the remote endpoints for tunnel interfaces.
show nsx controller	Displays connection status of the NSX Controller. Includes an option to display the gateway certificate that is needed for NSX "transport node" configuration.
show overlay-gateway	Displays status and statistics for the VXLAN overlay-gateway instance.
show running-config overlay-gateway	Displays the running configuration of the overlay gateway configuration, including the connection type.
show tunnel	Displays tunnel statistics, including those for the NSX Service Node.

Troubleshooting VXLAN Layer 2 extension configurations

This section provides examples of a variety of **show** commands that can be used to troubleshoot VXLAN Layer 2 extension configurations. In general, do the following to confirm the proper operation of the configuration:

- Confirm that the overlay gateway is up, with a valid VTEP IP address on all RBridges.
- Confirm that the tunnel interface is a member of the VLAN.
- Confirm that the administrative and operational states of the tunnels are up on all RBridges.
- Confirm that there is one multicast-replicator enabled RBridge for every extension tunnel.
- Confirm that only one BUM forwarder is enabled for the next hop for every extension tunnel.
- Confirm that all sites are configured so that the tunnels form a full mesh, and that there are no VLAN extension inconsistencies across sites.
- Confirm the MAC addresses.

Confirming the status of the VXLAN Layer 2 extension overlay gateway

NOTE

In the following examples, items to look for and confirm are highlighted in **bold.**

Confirm the IP address, loopback ID, and administrative state, by using the **show overlay-gateway** command:

```
sw0(config-overlay-gw-gw121)# do show overlay-gateway

Overlay Gateway "gw121", ID 1, rbridge-ids 1
Type Layer2-Extension, Tunnel mode VXLAN
IP address 10.2.2.1 ( Loopback 100 ), Vrf default-vrf
Admin state up
Number of tunnels 0
Packet count: RX 0               TX 0
Byte count  : RX (NA)            TX 0
```

NOTE

Ensure that there are no "Unknown" entries.

Confirming that the tunnel interface is a member of the VLAN

Confirm the tunnel and VNIs, by using the **show vlan brief** command:

```
switch# show vlan brief

Total Number of VLANs configured    : 14
Total Number of VLANs provisioned   : 14
Total Number of VLANs unprovisioned : 0
VLAN    Name     State                      Ports           Classification
(F)-FCoE                                    (u)-Untagged
(R)-RSPAN                                   (c)-Converged
(T)-TRANSPARENT                             (t)-Tagged
======= ======== ========================== =============== ==================
1       default  ACTIVE                     Po 333(t)
                                            Po 444(t)
1002(F) VLAN1002 INACTIVE(no member port)
5000    VLAN5000 INACTIVE(member port down) Tu 61441(t)     vni 15000
6000    VLAN6000 INACTIVE(member port down) Tu 61441(t)     vni 16000
                                            Tu 61442(t)     vni 16000
6001    VLAN6001 INACTIVE(member port down) Tu 61441(t)     vni 16001
                                            Tu 61442(t)     vni 16001
6002    VLAN6002 INACTIVE(member port down) Tu 61441(t)     vni 16002
                                            Tu 61442(t)     vni 16002
```

Confirming the status of the VTEP and tunnel

Confirm the administrative and operational state, by using the **show tunnel brief rbridge-id** command for the relevant RBridges:

```
switch# show tunnel brief rbridge-id 10

Tunnel 61441, mode VXLAN, rbridge-ids 10
Admin state UP, Oper state UP
Source IP 10.2.2.1, Vrf default-vrf
Destination IP 10.1.1.1


switch# show tunnel brief rbridge-id 40

Tunnel 61441, mode VXLAN, rbridge-ids 40
Admin state UP, Oper state UP
Source IP 10.2.2.1, Vrf default-vrf
Destination IP 10.1.1.1
```

NOTE

Ensure that there are no discrepancies in the operational states and source IP addresses across RBridges.

Confirming source and destination IP and MAC addresses

Confirm the source and destination IP and MAC addresses, by using the **show tunnel** command:

```
switch# show tunnel 61441

Tunnel 61441, mode VXLAN, rbridge-ids 10,40
Ifindex 2080374796, Admin state up, Oper state up
Overlay gateway "test", ID 1
Source IP 10.2.2.1 ( Loopback 100 ), Vrf default-vrf
Destination IP 10.1.1.1, Site VCS_2
Active next hops on rbridge 10:
    IP: 10.2.2.2, Vrf: default-vrf
    Egress L3 port: Ve 20, Outer SMAC: 0027.f86f.cc18
    Outer DMAC: 0011.bbbb.dddd
    Egress L2 Port: Po 10, Outer ctag: 20
    BUM forwarder: no
    IP: 10.3.3.3, Vrf: default-vrf
    Egress L3 port: Ve 30, Outer SMAC: 0027.f86f.2233
    Outer DMAC: 0011.bbbb.1111
    Egress L2 Port: Po 30, Outer ctag: 30
    BUM forwarder: yes

Active next hops on rbridge 40:
    IP: 10.1.1.1, Vrf: default-vrf
    Egress L3 port: Te 20/0/1, Outer SMAC: 0027.f86f.3344
    Outer DMAC: 0000.0000.dddd
    Egress L2 Port: Po 10, Outer ctag: 20
    BUM forwarder: no

Packet count: RX 0              TX 0
Byte count  : RX (NA)           TX 00
```

NOTE

You can also use the **rbridge-id** keyword. Watch out for discrepancies in the operational states and source and destination IP addresses across RBridges. The "Active next hops . . . " lines can vary across participating RBridges. Ensure that BUM forwarding is enabled for the next hop, which carries egress BUM traffic. There must be only one such next hop for the extension tunnel.

To view tunnel details for the site, you can use the **show tunnel site** command with the **brief** keyword:

```
switch# show tunnel site VCS_2 brief

Tunnel 61441, mode VXLAN, rbridge-ids 10
Admin state UP, Oper state UP
Source IP 10.2.2.1, Vrf default-vrf
Destination IP 10.1.1.1
```

Viewing tunnel statistics of the VXLAN Layer 2 extension overlay gateway

Confirm the transmission of packets on tunnels, by using the **show tunnel statistics** command:

NOTE

Receive bytes are not available.

```
switch# show tunnel statistics
Tnl ID   RX packets      TX packets      RX bytes        TX bytes
======== =============== =============== =============== ================
61441    123             456             (NA)            4560
61442    567             890             (NA)            8900
```

Confirming the MAC addresses for VLANs and tunnels

Confirm the MAC addresses and the state of the tunnels for the appropriate VLANs:

```
switch# show mac-address-table

VlanId  Mac-address    Type    State  Ports

30      0000.0000.0001 Dynamic Active Tu 61441
30      0000.0000.0002 Dynamic Active Tu 61441
30      0000.0000.0003 Dynamic Active Tu 61441
30      0000.0000.0004 Dynamic Active Tu 61441
30      0000.0000.0005 Dynamic Active Tu 61442
30      0000.0000.0006 Dynamic Active Tu 61442
20      0027.f880.1890 System  Remote XX 40/X/X
20      0027.f880.328f Dynamic Active Po 1
```

NOTE

Only an Active state confirms that the MAC address is getting forwarded locally unto the tunnel.

Viewing VXLAN overlay gateway statistics

Confirm transmit and receive packets for the VLAN, by using the **show overlay-gateway** command:

```
switch# show overlay-gateway name test vlan statistics

VLAN ID   RX packets           TX packets
========= ==================== ====================
30        1234                 5678
```

Confirming the tunnel SPAN status

Confirm the session status and the source and destination addresses by using the **show monitor** command:

```
switch# show monitor

Session                : 10
Description            : [None]
State                  : Enabled
Source Interface       : Tu 61442 (Down)
Destination Interface  : Te 1/0/48 (Up)
Direction              : Both
```

Confirm all sFlow details by using the **show sflow all** command:

```
switch# show sflow all

sFlow services are:                      enabled
Global default sampling rate:            32768 pkts
Global default counter polling interval: 20 secs
Rbridge-Id                          Collector server address                    Number of samples sent
-------------------------------------------------------------------------------------------------------
         2                                  1.1.1.1:6343                                          1212
sflow info for tunnel ifindex: 0x7c00f001
------------------------------
sFlow services are:           enabled
Samples received from hardware:       1212
Effective sample-rate:                512
```

Troubleshooting Virtual Fabrics

Use the following **show** commands to view the status of Virtual Fabric configurations. For details, refer to the *Network OS Command Reference*.

TABLE 16 Virtual Fabrics show commands

Command	Description
show vlan brief	Displays all classified VLANs that are configured, provisioned (active) or unprovisioned (inactive).
show port profile	Displays the AMPP port-profile configuration information.
show overlapping-vlan-resource usage	For VDX 6740 and VDX 6940 series only. Displays the utilization of the hardware table entries that support classified or transport VLAN classifications that use overlapping C-TAGs in a Virtual Fabric context.
show virtual-fabric status	Displays the status of the Virtual Fabric: VF-capable, VF-incapable, or VF-enabled.

Configuring CML with Virtual Fabrics

Beginning with Network OS release 5.0.0, support is added for conversational MAC learning (CML) on classified VLANs as well as on 802.1Q VLANs. For an overview and applicable configuration examples, refer to Conversational MAC learning on page 79.

STP-Type Protocols

STP overview

A network topology of bridges typically contains redundant connections to provide alternate paths in case of link failures. However, because there is no concept of TTL in Ethernet frames, this could result in the permanent circulation of frames if there are loops in the network. To prevent loops, a spanning tree connecting all the bridges is formed in real time. The redundant ports are put in a blocking (nonforwarding) state. They are enabled when required. In order to build a spanning tree for the bridge topology, the bridges must exchange control frames (BPDUs - Bridge Protocol Data Units). The protocols define the semantics of the BPDUs and the required state machine. The first Spanning Tree Protocol (STP) became part of the IEEE 802.1d standard.

The STP interface states for every Layer 2 interface running STP are as follows:

- *Blocking* — The interface does not forward frames.
- *Listening* — The interface is identified by the spanning tree as one that should participate in frame forwarding. This is a transitional state after the blocking state.
- *Learning* — The interface prepares to participate in frame forwarding.
- *Forwarding* — The interface forwards frames.
- *Disabled* — The interface is not participating in spanning tree because of a shutdown port, no link on the port, or no spanning tree instance running on the port.

A port participating in spanning tree moves through these states:

- From initialization to blocking
- From blocking to listening or to disabled
- From listening to learning or to disabled
- From learning to forwarding, blocking, or disabled
- From forwarding to disabled

The following STP features are considered optional features, although you might use them in your STP configuration:

- *Root guard*
- *Port fast BPDU guard* and *BPDU filter*

STP configuration guidelines and restrictions

Follow these configuration guidelines and restrictions when configuring STP:

- You have to disable one form of xSTP before enabling another.
- Packet drops or packet flooding may occur if you do not enable xSTP on all devices connected on both sides of parallel links.
- Network OS switches in logical chassis cluster mode or fabric cluster mode drop all tagged xSTP frames that originate from the Brocade MLX-MCT.
- In the case of STP over VCS (STPoVCS), STP is disabled by default on all ports.

- When a misconfigured local area network running spanning tree has one or more loops, a traffic storm of spanning tree BPDUs can occur. In certain circumstances, VDX switches can reboot when subjected to an extended period of traffic storm involving spanning tree BPDUs.
- Additionally, when a misconfigured local area network running spanning tree has one or more loops, a traffic storm of spanning tree BPDUs can occur. Edge Loop Detection (ELD) protocol cannot eliminate loops during a traffic storm involving control packets, such as spanning tree BPDUs.
- Do not force an alternate root path through root path cost with PVST+ or R-PVST+ on legacy Foundry equipment, such as the Brocade NetIron MLX or Brocade TurboIron. This can cause traffic issues on the network.
- For load balancing across redundant paths in the network to work, all VLAN-to-instance mapping assignments must match; otherwise, all traffic flows on a single link.
- Spanning tree topologies must not be enabled on any direct server connections to the front-end 10-gigabit Ethernet ports that may run FCoE traffic. This may result in lost or dropped FCoE logins.

RSTP

NOTE
Rapid Spanning Tree Protocol is designed to be compatible and interoperate with STP. However, the advantages of the RSTP fast reconvergence are lost when it interoperates with switches running STP.

The IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard is an evolution of the 802.1D STP standard. It provides rapid reconvergence following the failure of a switch, a switch port, or a LAN. It provides rapid reconvergence of edge ports, new root ports, and ports connected through point-to-point links.

The RSTP interface states for every Layer 2 interface running RSTP are as follows:

- *Learning* — The interface prepares to participate in frame forwarding.
- *Forwarding* — The interface forwards frames.
- *Discarding* — The interface discards frames. Note that the 802.1D disabled, blocking, and listening states are merged into the RSTP discarding state. Ports in the discarding state do not take part in the active topology and do not learn MAC addresses.

The following table lists the interface state changes between STP and RSTP.

TABLE 17 STP versus RSTP state comparison

STP interface state	RSTP interface state	Is the interface included in the active topology?	Is the interface learning MAC addresses?
Disabled	Discarding	No	No
Blocking	Discarding	No	No
Listening	Discarding	Yes	No
Learning	Learning	Yes	Yes
Forwarding	Forwarding	Yes	Yes

With RSTP, the port roles for the interface are also different. RSTP differentiates explicitly between the state of the port and the role it plays in the topology. RSTP uses the root port and designated port roles defined by STP, but splits the blocked port role into backup port and alternate port roles:

- *Backup port* — Provides a backup for the designated port and can only exist where two or more ports of the switch are connected to the same LAN; the LAN where the bridge serves as a designated switch.
- *Alternate port* — Serves as an alternate port for the root port providing a redundant path towards the root bridge.

Only the root port and the designated ports are part of the active topology; the alternate and backup ports do not participate in it.

When the network is stable, the root and the designated ports are in the forwarding state, while the alternate and backup ports are in the discarding state. When there is a topology change, the new RSTP port roles allow a faster transition of an alternate port into the forwarding state.

MSTP

IEEE 802.1s Multiple STP (MSTP) helps create multiple loop-free active topologies on a single physical topology. MSTP enables multiple VLANs to be mapped to the same spanning tree instance (forwarding path), which reduces the number of spanning tree instances needed to support a large number of VLANs. Each MSTP instance has a spanning tree topology independent of other spanning tree instances. With MSTP you can have multiple forwarding paths for data traffic. A failure in one instance does not affect other instances. With MSTP, you are able to more effectively utilize the physical resources present in the network and achieve better load balancing of VLAN traffic.

> NOTE
> In MSTP mode, RSTP is automatically enabled to provide rapid convergence.

Multiple switches must be configured consistently with the same MSTP configuration to participate in multiple spanning tree instances. A group of interconnected switches that have the same MSTP configuration is called an MSTP region.

> NOTE
> Brocade supports 32 MSTP instances and one MSTP region.

MSTP introduces a hierarchical way of managing switch domains using regions. Switches that share common MSTP configuration attributes belong to a region. The MSTP configuration determines the MSTP region where each switch resides. The common MSTP configuration attributes are as follows:

- Alphanumeric configuration name (32 bytes)
- Configuration revision number (2 bytes)
- 4096-element table that maps each of the VLANs to an MSTP instance

Region boundaries are determined by the above configuration. A multiple spanning tree instance is an RSTP instance that operates inside an MSTP region and determines the active topology for the set of VLANs mapping to that instance. Every region has a common internal spanning tree (CIST) that forms a single spanning tree instance that includes all the switches in the region. The difference between the CIST instance and the MSTP instance is that the CIST instance operates across the MSTP region and forms a loop-free topology across regions, while the MSTP instance operates only within a region. The CIST instance can operate using RSTP if all the switches across the regions support RSTP. However, if any of the switches operate using 802.1D STP, the CIST instance reverts to 802.1D. Each region is viewed logically as a single STP/RSTP bridge to other regions.

MSTP guidelines and restrictions

Follow these restrictions and guidelines when configuring MSTP:

- You can have 32 MSTP instances and one MSTP region.
- Create VLANs before mapping them to MSTP instances.
- The MSTP **force-version** option is not supported.
- When you enable MSTP by using the **global protocol spanning-tree mstp** command, RSTP is enabled automatically.
- For two or more switches to be in the same MSTP region, they must have the same VLAN-to-instance map, the same configuration revision number, and the same name.

PVST+ and Rapid PVST+

Both STP and RSTP build a single logical topology. A typical network has multiple VLANs. A single logical topology does not efficiently utilize the availability of redundant paths for multiple VLANs. If a port is set to "blocked/discarding" for one VLAN (under STP/RSTP), it is the same for all other VLANs too.

Per-VLAN Spanning Tree Plus (PVST+) protocol runs a spanning tree instance for each VLAN in the network. The version of PVST+ that uses the RSTP state machine is called Rapid-PVST Plus (R-PVST+). R-PVST+ has one instance of spanning tree for each VLAN on the switch.

PVST+ is not a scalable model when there are many VLANs in the network, as it consumes a lot of CPU power. A reasonable compromise between the two extremes of RSTP and R-PVST+ is the Multiple Spanning Tree protocol (MSTP), which was standardized as IEEE 802.1s and later incorporated into the IEEE 802.1Q-2003 standard. MSTP runs multiple instances of spanning tree that are independent of VLANs. It then maps a set of VLANs to each instance.

> NOTE
> Network OS 4.0 and later supports PVST+ and R-PVST+ only. The PVST and R-PVST protocols are proprietary to Cisco and are not supported.

PVST+ and R-PVST+ guidelines and restrictions

Consider the following when configuring PVST+ and R-PVST+:

- Disabling the tagging of native VLANs is required on edge ports in fabric cluster mode; otherwise, PVST+/R-PVST+ does not converge and forms a loop on the native VLAN. The tagged native VLAN data traffic is ignored. The native VLAN untagged data is forwarded.
- Disabling the tagging of native VLANs is required on STP/RSTP/MSTP VDX devices in standalone mode, otherwise PVST/RPVST does not converge and forms a loop on the native VLAN. The tagged native VLAN data traffic is ignored. The native VLAN untagged data is forwarded.
- If a VLAN is configured with tagged ports that have RSTP mode enabled but not PVST+ mode enabled, and are connected to VDX devices, then BPDUs from the tagged ports that are received by the VDX device are dropped.

Spanning Tree Protocol and VCS mode

Network OS 4.0 and later supports any version of STP to run in VCS mode and function correctly between interconnecting VCS nodes, or between VCS and other vendor's switches. This feature is called Distributed Spanning Tree Protocol (DiST).

The purpose of DiST is as follows:

- To support VCS to VCS connectivity and automatic loop detection and prevention.
- To assist deployment plans for replacing the legacy xSTP enabled switches in your network.

DiST supports any of the following flavors of xSTP:

- IEEE STP
- IEEE RSTP
- IEEE MSTP
- PVST
- PVRST

DiST treats one VCS as one virtual xSTP bridge from an external view. Each VCS has a unique RBridge ID and Priority. DiST can be enabled on VCS edge ports connecting to other VCS nodes. The Port Ids used for xSTP are dynamically assigned and unique within the VCS.

It is important to note that in fabric cluster mode, it is assumed that the global xSTP configuration is the same on all the member nodes of the VCS. Mismatched global configurations of xSTP across different nodes in VCS are not supported. For example, if one of the nodes in a VCS is configured for RSTP and another one is configured for STP or MSTP, this scenario is unsupported and the fabric will not form.

Each RBridge runs the spanning tree instance in a distributed manner. Each spanning tree instance considers all the edge ports and the best information from the remote RBridges to arrive at the spanning tree topology. Each RBridge updates all the other members about its best information for a given spanning tree instance.

Each RBridge maintains a table of best information from the other RBridges in the cluster. This table is identical across all the RBridges in the cluster. This information is used to derive the port roles for the local edge ports. The shared information of the whole cluster is considered in the spanning tree calculations for port roles and states of local edge ports. Thus, all the remote RBridges' edge port information could affect the port role selection and port state transitions for the local edge ports. This ensures that each RBridge considers the port roles and states of all the other RBridges to arrive at a final spanning tree topology.

In the event of a change of the "best" information on any member RBridge, that RBridge would update its own next best information to the other RBridges. Some of the scenarios in which this could happen are the following:

- Operational status change of port associated with the "best" information
- Reception of superior information by another edge port on the RBridge
- Reception of superior or inferior information by the "best" port on the RBridge
- Nonreception of BPDUs on the best port for a given period of time

The xSTP update information is received by all member nodes of the cluster. Each node updates its internal database with the received information. If this results in a best-information change, the update is applied on to the logical port for the node. This triggers the xSTP state machine for all local ports.

Configuring and managing STP and STP variants

Before proceeding, refer to STP configuration guidelines and restrictions on page 147.

Understanding the default STP configuration

It is helpful to understand the STP defaults before you make configuration changes. The following table lists the default STP configuration.

TABLE 18 Default STP configuration

Parameter	Default setting
Spanning-tree mode	By default, STP, RSTP, and MSTP are disabled
Bridge priority	32768
Bridge forward delay	15 seconds
Bridge maximum aging time	20 seconds
Error disable timeout timer	Disabled
Error disable timeout interval	300 seconds
Port-channel path cost	Standard
Bridge hello time	2 seconds

The following table lists those switch defaults which apply only to MSTP configurations.

TABLE 19 Default MSTP configuration

Parameter	Default setting
Cisco interoperability	Disabled
Switch priority (when mapping a VLAN to an MSTP instance)	32768
Maximum hops	20 hops
Revision number	0

The following table lists the switch defaults for the 10-gigabit Ethernet DCB interface-specific configuration.

TABLE 20 Default 10-gigabit Ethernet DCB interface-specific configuration

Parameter	Default setting
Spanning tree	Disabled on the interface
Automatic edge detection	Disabled
Path cost	2000
Edge port	Disabled
Guard root	Disabled
Hello time	2 seconds
Link type	Point-to-point
Port fast	Disabled
Port priority	128
DCB interface root port	Allow the DCB interface to become a root port.
DCB interface BPDU restriction	Restriction is disabled.

Saving configuration changes

Enter the **copy running-config startup-config** command to save your configuration changes.

Configuring STP

The process for configuring STP is as follows:

1. Enter global configuration mode.
2. Enable STP by using the global **protocol spanning-tree** command.

```
switch(config)# protocol spanning-tree stp
```

3. Designate the root switch by using the **bridge-priority** command. The range is 0 through 61440 and the priority values can be set only in increments of 4096.

```
switch(conf-stp)# bridge-priority 28672
```

4. Enable port fast on switch ports by using the **spanning-tree portfast** command.

ATTENTION
Note the following conditions:

- Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat these commands for every port connected to workstations or PCs. Do not enable port fast on ports that connect to other switches.

- If BPDUs are received on a port fast enabled interface, the interface loses the edge port status unless it receives a shut/no shut.
- Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.

```
switch(config)# interface tengigabitethernet 1/0/10
switch(conf-if-te-1/0/10)# spanning-tree portfast
switch(conf-if-te-1/0/10)# exit
```

5. To interoperate with non-VDX devices (such as NetIron and FastIron) in PVST+/R-PVST+ mode, you may need to configure the interface that is connected to that switch by using the **spanning-tree bpdu-mac** command.

```
switch(config)# interface tengigabitethernet 1/0/12
switch(conf-if-te-0/12)# spanning-tree bpdu-mac 0100.0ccc.cccd
```

6. Specify port priorities by using the **spanning-tree priority** command to influence the selection of root/designated ports.
7. Return to privileged EXEC mode.

```
switch(conf-if-te-1/0/12)# end
```

8. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

When the spanning tree topology is completed, the network switches send and receive data only on the ports that are part of the spanning tree. Data received on ports that are not part of the spanning tree is blocked.

NOTE
Brocade recommends leaving other STP variables at their default values.

Configuring RSTP

The process for configuring RSTP is as follows.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enable RSTP by using the global **protocol spanning-tree** command.

```
switch(config)# protocol spanning-tree rstp
```

3. Designate the root switch by using the **bridge-priority** command. The range is 0 through 61440 and the priority values can be set only in increments of 4096.

```
switch(conf-stp)# bridge-priority 28582
```

4. Configure the bridge forward delay value.

```
switch(conf-stp)# forward-delay 20
```

5. Configure the bridge maximum aging time value.

```
switch(conf-stp)# max-age 25
```

6. Enable the error-disable-timeout timer.

```
switch(conf-stp)# error-disable-timeout enable
```

7. Configure the error-disable-timeout interval value.

```
switch(conf-stp)# error-disable-timeout interval 60
```

8. Configure the port-channel path cost.

```
switch(conf-stp)# port-channel path-cost custom
```

9. Configure the bridge hello-time value.

```
switch(conf-stp)# hello-time 5
```

10. Enable port fast on switch ports by using the **spanning-tree portfast** command.

NOTE
Port fast only needs to be enabled on ports that connect to workstations or PCs. Repeat these commands for every port connected to workstations or PCs. Do not enable port fast on ports that connect to other switches.

NOTE
Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.

```
switch(config)# interface tengigabitethernet 1/0/10
switch(conf-if-te-1/0/10)# spanning-tree portfast
```

11. Specify port priorities by using the **spanning-tree priority** command to influence the selection of root/designated ports.
12. Return to privileged EXEC mode.

```
switch(config)# end
```

13. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Configuring MSTP

The process for configuring MSTP is as follows.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enable MSTP by using the global **protocol spanning-tree** command.

```
switch(config)# protocol spanning-tree mstp
```

3. Specify the region name by using the **region** *region_name* command.

```
switch(config-mstp)# region brocade1
```

4. Specify the revision number by using the **revision** command.

```
switch(config-mstp)# revision 1
```

5. Map a VLAN to an MSTP instance by using the **instance** command.

```
switch(config-mstp)# instance 1 vlan 2, 3
switch(config-mstp)# instance 2 vlan 4-6
switch(config-mstp)# instance 1 priority 4096
```

6. Specify the maximum hops for a BPDU to prevent the messages from looping indefinitely on the interface by using the **max-hops** *hop_count* command.

```
switch(config-mstp)# max-hops 25
```

7. Return to privileged EXEC mode.

```
switch(config)# end
```

8. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Configuring additional MSTP parameters

The following sections discuss how to configure additional MSTP parameters.

Enabling and disabling Cisco interoperability (MSTP)

In MSTP mode, use the **cisco-interoperability** command to enable or disable the ability to interoperate with certain legacy Cisco switches. If Cisco interoperability is required on any switch in the network, then all switches in the network must be compatible, and therefore enabled by means of this command. The default is Cisco interoperability is disabled.

> NOTE
> The **cisco-interoperability** command is necessary because the "version 3 length" field in the MSTP BPDU on some legacy Cisco switches does not conform to current standards.

To enable interoperability with certain legacy Cisco switches, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **cisco-interoperability enable** command to enable interoperability with certain legacy Cisco switches.

```
switch(config-mstp)# cisco-interoperability enable
```

4. (Optional) To disable interoperability with certain legacy Cisco switches, enter the **cisco-interoperability disable** command.

```
switch(config-mstp)# cisco-interoperability disable
```

Mapping a VLAN to an MSTP instance

In MSTP mode, use the **instance** command to map a VLAN to an MTSP. You can group a set of VLANs to an instance. This command can be used only after the VLAN is created. VLAN instance mapping is removed from the configuration if the underlying VLANs are deleted.

To map a VLAN to an MSTP instance, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Map a VLAN to an MSTP instance.

```
switch(config-mstp)# instance 5 vlan 300
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

5. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Specifying the maximum number of hops for a BPDU (MSTP)

In MSTP mode, use the **max-hops** command to configure the maximum number of hops for a BPDU in an MSTP region. Specifying the maximum hops for a BPDU prevents the messages from looping indefinitely on the interface. When you change the number of hops, it affects all spanning tree instances. The range is 1 through 40. The default is 20 hops.

To configure the maximum number of hops for a BPDU in an MSTP region, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **max-hops 30** command to change the maximum number of hops from the default for a BPDU in an MSTP region.

```
switch(config-mstp)# max-hops 30
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

5. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Specifying a name for an MSTP region

In MSTP mode, use the **region** command to assign a name to an MSTP region. The region name has a maximum length of 32 characters and is case-sensitive.

To assign a name to an MSTP region, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.

```
switch# config
```

2. Enter the **protocol** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **region** command to assign a name to an MSTP region.

```
switch(config-mstp)# region sydney
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

5. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Specifying a revision number for MSTP configuration

In MSTP mode, use the **revision** command to specify a revision number for an MSTP configuration. The range is 0 through 255. The default is 0.

To specify a revision number for an MSTP configuration, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree mstp** command to enable MSTP.

```
switch(config)# protocol spanning-tree mstp
```

3. Enter the **revision** command to specify a revision number for an MSTP configuration.

```
switch(config-mstp)# revision 17
```

4. Return to privileged EXEC mode.

```
switch(config-mstp)# end
```

5. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Configuring PVST+ or R-PVST+

To configure PVST+ or R-PVST+, use the **protocol spanning-tree pvst** and **protocol spanning-tree rpvst** commands. Refer to the *Network OS Command Reference* for details.

The following example script configures PVST+:

```
switch(config)# protocol spanning-tree pvst
switch(conf-pvst)# bridge-priority 4096
switch(conf-pvst)# forward-delay 4
switch(conf-pvst)# hello-time 2
switch(conf-pvst)# max-age 7
```

The following example script configures R-PVST+:

```
switch(config)# protocol spanning-tree rpvst
switch(conf-pvst)# bridge-priority 4096
switch(conf-pvst)# forward-delay 4
switch(conf-pvst)# hello-time 2
switch(conf-pvst)# max-age 7
```

Enabling STP, RSTP, MSTP, PVST+ or R-PVST+

Enable STP to detect and avoid loops. By default, STP, RSTP, MSTP, PVST+, or R-PVST+ are not enabled. You must turn off one form of STP before turning on another form.

Perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** *mode_name* command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree pvst
```

To disable STP, RSTP, MSTP, PVST+, or R-PVST+, enter the **no protocol spanning-tree** command:

```
switch(config)# no protocol spanning-tree
```

NOTE
The above command deletes the context and all the configurations defined within the context or protocol for an interface.

Shutting down STP, RSTP, MSTP, PVST+, or R-PVST+ globally

To shut down STP, RSTP, MSTP, PVST+, or R-PVST+ globally, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **shutdown** command to shut down STP, RSTP, MSTP, PVST+, or R-PVST+ globally. The **shutdown** command works in all modes.

```
switch(config-mstp)# shutdown
```

Specifying bridge parameters

There are a variety of options for configuring the behavior of the STP bridging function, as shown in the following subsections.

Specifying the bridge priority

In PVST+ or R-PVST+ mode, use the **bridge-priority** command to specify the priority of the switch. After you decide on the root switch, set the appropriate values to designate the switch as the root switch. If a switch has a bridge priority that is lower than that of all the other switches, the other switches automatically select the switch as the root switch.

The root switch should be centrally located and not in a "disruptive" location. Backbone switches typically serve as the root switch because they often do not connect to end stations. All other decisions in the network, such as which port to block and which port to put in forwarding mode, are made from the perspective of the root switch.

Bridge Protocol Data Units (BPDUs) carry the information exchanged between switches. When all the switches in the network are powered up, they start the process of selecting the root switch. Each switch transmits a BPDU to directly connected switches on a per-VLAN basis. Each switch compares the received BPDU to the BPDU that the switch sent. In the root switch selection process, if switch 1 advertises a root ID that is a lower number than the root ID that switch 2 advertises, switch 2 stops the advertisement of its root ID, and accepts the root ID of switch 1. The switch with the lowest bridge priority becomes the root switch.

Additionally, you may specify the bridge priority for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To specify the bridge priority, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable PVST+ or R-PVST+.

```
switch(config)# protocol spanning-tree pvst
```

3. Specify the bridge priority. The range is 0 through 61440 and the priority values can be set only in increments of 4096. The default priority is 32678.

```
switch(conf-stp)# bridge-priority 20480
```

4. Specify the bridge priority for a specific VLAN.

```
switch(conf-stp)# bridge-priority 20480 vlan 10
```

Specifying the bridge forward delay

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **forward-delay** command to specify how long an interface remains in the listening and learning states before the interface begins forwarding all spanning tree instances. The valid range is from 4 through 30 seconds. The default is 15 seconds. The following relationship should be kept:

```
2*(forward_delay - 1)≥ max_age ≥ 2*(hello_time + 1)
```

Additionally, you may specify the forward delay for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To specify the bridge forward delay, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the bridge forward delay.

```
switch(conf-stp)# forward-delay 20
```

4. Specify the bridge forward delay for a specific VLAN.

```
switch(conf-stp)# forward-delay 20 vlan 10
```

Specifying the bridge maximum aging time

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **max-age** command to control the maximum length of time that passes before an interface saves its BPDU configuration information.

When configuring the maximum aging time, you must set the max-age to be greater than the hello time. The range is 6 through 40 seconds. The default is 20 seconds. The following relationship should be kept:

```
2*(forward_delay - 1) ≥ max_age ≥ 2*(hello_time + 1)
```

Additionally, you may specify the maximum aging for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To specify the bridge maximum aging time, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the bridge maximum aging time.

```
switch(conf-stp)# max-age 25
```

4. (Optional) Specify the bridge maximum aging time for a specific VLAN.

```
switch(conf-stp)# max-age 25 vlan 10
```

Specifying the bridge hello time

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **hello-time** command to configure the bridge hello time. The hello time determines how often the switch interface broadcasts hello BPDUs to other devices. The range is from 1 through 10 seconds. The default is 2 seconds.

When configuring the hello time, you must set the maximum age to be greater than the hello time. The following relationship should be kept:

```
2*(forward_delay - 1) ≥ max_age ≥ 2*(hello_time + 1)
```

Additionally, you may specify the hello time for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To specify the bridge hello time, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# configure terminal
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the time range in seconds for the interval between the hello BPDUs sent on an interface.

```
switch(conf-stp)# hello-time 5
```

4. (Optional) Specify the time range in seconds for the interval between the hello BPDUs sent on an interface for a specific VLAN.

```
switch(conf-stp)# hello-time 5 vlan 10
```

5. Return to privileged EXEC mode.

```
switch(config)# end
```

6. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Configuring STP timers

You must configure the error disable timeout timer before you can use it.

Enabling the error disable timeout timer

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **error-disable-timeout** command to enable a timer that will bring a port out of the disabled state. When the STP BPDU guard disables a port, the port remains in the disabled state unless the port is enabled manually. This command allows you to enable the port from the disabled state. For details on configuring the error disable timeout interval, refer to Specifying the error disable timeout interval on page 161.

To enable the error disable timeout timer, perform the following steps from privileged EXEC mode. By default, the timeout feature is disabled.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Enable the error disable timeout timer.

```
switch(conf-stp)# error-disable-timeout enable
```

Specifying the error disable timeout interval

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **error-disable-timeout** command to specify the time in seconds it takes for an interface to time out. The range is from 10 through 1000000 seconds. The default is 300 seconds. By default, the timeout feature is disabled.

To specify the time in seconds it takes for an interface to time out, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the time in seconds it takes for an interface to time out.

```
switch(conf-stp)# error-disable-timeout interval 60
```

Specifying the port-channel path cost

In STP, RSTP, MSTP, PVST+, or R-PVST+ mode, use the **port-channel path-cost** command to specify the port-channel path cost. The default port cost is **standard**. The path cost options are as follows:

- **custom** — Specifies that the path cost changes according to the port-channel's bandwidth.
- **standard** — Specifies that the path cost does not change according to the port-channel's bandwidth.

To specify the port-channel path cost, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Enter the **protocol spanning-tree** command to enable STP, RSTP, MSTP, PVST+, or R-PVST+.

```
switch(config)# protocol spanning-tree stp
```

3. Specify the port-channel path cost.

```
switch(conf-stp)# port-channel path-cost custom
```

4. Return to privileged EXEC mode.

```
switch(config)# end
```

5. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Specifying the transmit hold count (RSTP, MSTP, and R-PVST+)

In RSTP, MSTP, or R-PVST+ mode, use the **transmit-holdcount** command to configure the BPDU burst size by specifying the transmit hold count value. The command configures the maximum number of BPDUs transmitted per second for RSTP and MSTP before pausing for 1 second. The range is 1 through 10. The default is 6.

To specify the transmit hold count, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to change to global configuration mode.

```
switch# config
```

2. Specify the transmit hold count.

```
switch(config-mstp)# transmit-holdcount 5
```

3. Return to privileged EXEC mode.

```
switch(config)# end
```

4. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Clearing spanning tree counters

To clear spanning tree counters on all interfaces or on the specified interface, use the **clear spanning-tree counter** command in privileged EXEC mode.

1. Use the **clear spanning-tree counter** command to clear the spanning tree counters on all the interfaces.

```
switch# clear spanning-tree counter
```

2. Alternatively, use the **clear spanning-tree counter interface** command to clear the spanning tree counters associated with a specific port-channel or DCB port interface.

```
switch# clear spanning-tree counter interface tengigabitethernet 1/0/1
```

Clearing spanning tree-detected protocols

To restart the protocol migration process (force the renegotiation with neighboring switches) on either all interfaces or on a specified interface, use the **clear spanning-tree detected-protocols** command in privileged EXEC mode.

To restart the protocol migration process, perform either of the following tasks:

1. Use the **clear spanning-tree detected-protocols** command to clear all spanning tree counters on all interfaces:

```
switch# clear spanning-tree detected-protocols
```

2. Alternatively, use the **clear spanning-tree detected-protocols interface** *interface_ID* command to clear the spanning tree counters associated with a specific port-channel or DCB port interface:

```
switch# clear spanning-tree detected-protocols interface tengigabitethernet 0/1
```

Displaying STP, RSTP, MSTP, PVST+, or R-PVST+ information

Enter the **show spanning-tree brief** command in privileged EXEC mode to display all STP, RSTP, MSTP, PVST+, or R-PVST+-related information.

NOTE
The **show spanning-tree brief** command output shows the port state as ERR, not root_inc, when **root guard** is in effect.

Configuring STP, RSTP, or MSTP on DCB interface ports

By default, STP is not enabled on individual ports. This section details the commands for enabling and configuring STP, RSTP, or MSTP on individual 10-gigabit Ethernet Data Center Bridging (DCB) interface ports.

Enabling and disabling STP (DCB)

Do the following to enable or disable spanning tree, as appropriate.

Enabling spanning tree (DCB)

From the DCB interface, use this command to enable spanning tree on the DCB interface. By default, spanning tree is disabled.

To enable spanning tree on the DCB interface, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **no spanning-tree shutdown** command to enable spanning tree on the DCB interface.

```
switch(conf-if-te-1/0/1)# no spanning-tree shutdown
```

Disabling spanning tree (DCB)

From the DCB interface, use the **spanning-tree shutdown** command to disable spanning tree on the DCB interface. By default, spanning tree is disabled.

To disable spanning tree on the DCB interface, perform the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree shutdown** command to disable spanning tree on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree shutdown
```

Enabling automatic edge detection (DCB)

From the DCB interface, use the **spanning-tree autoedge** command to identify the edge port automatically. The port can become an edge port if no BPDU is received. By default, automatic edge detection is disabled.

To enable automatic edge detection on the DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree autoedge** command to enable automatic edge detection on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree autoedge
```

Configuring the path cost (DCB)

From the DCB interface, use the **spanning-tree cost** command to configure the path cost for spanning tree calculations. The lower the path cost means there is a greater chance of the interface becoming the root port. The range is 1 through 200000000. The default path cost is 2000 for a 10-gigabit Ethernet interface.

Additionally, you may specify the spanning tree cost for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To configure the path cost for spanning tree calculations on the DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree cost** command to configure the path cost for spanning tree calculations on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree cost 10000
```

5. Enter the **spanning-tree vlan** command to configure the path cost for spanning tree calculations on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree vlan 10 cost 10000
```

6. Return to privileged EXEC mode.

```
switch(conf-if-te-1/0/1)# end
```

7. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Enabling a port (interface) as an edge port (DCB)

From the DCB interface, use the **spanning-tree edgeport** command to enable the port as an edge port to allow the port to quickly transition to the forwarding state.

> NOTE
> The **spanning-tree edgeport** command is only for RSTP and MSTP. Use the **spanning-tree portfast** command for STP (refer to Enabling port fast (DCB) on page 168).

Follow these guidelines to configure a port as an edge port:

- A port can become an edge port if no BPDU is received.
- When an edge port receives a BPDU, it becomes a normal spanning tree port and is no longer an edge port.
- Because ports that are directly connected to end stations cannot create bridging loops in the network, edge ports transition directly to the forwarding state and skip the listening and learning states.

To enable the DCB interface as an edge port, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree edgeport** command to enable the DCB interface as an edge port.

```
switch(conf-if-te-1/0/1)# spanning-tree edgeport
```

Enabling guard root (DCB)

From the DCB interface, use this command to enable the guard root feature on the switch. This feature provides a way to enforce the root bridge placement in the network. With guard root enabled on an interface, the switch is able to restrict which interface is allowed to be the spanning tree root port or the path to the root for the switch. The root port provides the best path from the switch to the root switch. By default, guard root is disabled.

Guard root protects the root bridge from malicious attacks and unintentional misconfigurations where a bridge device that is not intended to be the root bridge becomes the root bridge. This causes severe bottlenecks in the data path. Guard root ensures that the port on which it is enabled is a designated port. If the guard-root-enabled port receives a superior BPDU, it goes to a discarding state.

Additionally, you may enable guard root for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To enable guard root on a DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree guard root** command to enable guard root on a DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree guard root
```

5. Optionally, enter the **spanning-tree** command to enable guard root for a specific VLAN.

```
switch(conf-if-te-1/0/1)# spanning-tree guard root vlan 10
```

Specifying the STP hello time (DCB)

From the DCB interface, use **spanning-tree hello-time** command to set the time interval between BPDUs sent by the root switch. Changing the hello time affects all spanning tree instances.

The maximum age setting must be greater than the hello time setting (refer to Specifying the bridge maximum aging time on page 159). The range for the **spanning-tree hello-time** command is 1 through 10 seconds. The default value is 2 seconds.

To specify the MSTP hello time on a DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree** command to specify the hello time on a DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree hello-time 5
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-1/0/1)# end
```

6. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Specifying restrictions for an MSTP instance (DCB)

From the DCB interface, use the **spanning-tree instance** command to specify restrictions on the interface for an MSTP instance.

To specify restrictions for an MSTP instance on a DCB interface, perform the following steps.

1. Enter the **config** command to access global configuration mode from privileged EXEC mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

   The **gigabitethernet** *rbridge-id/slot/port* operand is used only for the Brocade VDX 6710, Brocade VDX 8770-4, and Brocade VDX 8770-8. The prompt for these ports is in the following example format: switch(config-if-gi-22/0/1)#

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree instance restricted-tcn** command to restrict Topology Change Notification (TCN) BPDUs for an MSTP instance on a DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree instance 5 restricted-tcn
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-1/0/1)# end
```

6. Enter the **copy running-config startup-config** command to save the running configuration to the startup configuration.

```
switch# copy running-config startup-config
```

Specifying a link type (DCB)

From the DCB interface, use the **spanning-tree link-type** command to specify a link type. Specifying the **point-to-point** keyword enables rapid spanning tree transitions to the forwarding state. Specifying the **shared** keyword disables spanning tree rapid transitions. The default setting is point-to-point.

To specify a link type on a DCB interface, perform the following steps from privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree link-type shared** command, as in the following example, to change the link type from the default.

```
switch(conf-if-te-1/0/1)# spanning-tree link-type shared
```

Enabling port fast (DCB)

From the DCB interface, use the **spanning-tree portfast** command to enable port fast on an interface to allow the interface to transition quickly to the forwarding state. Port fast immediately puts the interface into the forwarding state without having to wait for the standard forward time.

NOTE

If you enable the **portfast bpdu-guard** option on an interface and the interface receives a BPDU, the software disables the interface and puts the interface in the ERR_DISABLE state.

CAUTION

Enabling port fast on ports can cause temporary bridging loops, in both trunking and nontrunking mode.

Use the **spanning-tree edgeport** command for MSTP, RSTP, and R-PVST+ (refer to Enabling a port (interface) as an edge port (DCB) on page 165).

To enable port fast on the DCB interface for STP, perform the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-0/1)# no shutdown
```

4. Enter the **spanning-tree portfast** command to enable port fast on the DCB interface.

```
switch(conf-if-te-0/1)# spanning-tree portfast
```

Specifying the port priority (DCB)

From the DCB interface, use the **spanning-tree priority** command to specify the port priority. The range is from 0 through 240 in increments of 16. The default value is 128.

In addition, you may specify the spanning tree priority for a specific VLAN. If the VLAN parameter is not provided, the priority value is applied globally for all per-VLAN instances. However, for the VLANs that have been configured explicitly, the per-VLAN configuration takes precedence over the global configuration.

To specify the port priority on the DCB interface, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree** command to specify the port priority on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree priority 32
```

5. (Optional) Enter the **spanning-tree vlan priority** command to specify the port priority for a specific VLAN.

```
switch(conf-if-te-1/0/1)# spanning-tree vlan 10 priority 32
```

Restricting the port from becoming a root port (DCB)

From the DCB interface, use the **spanning-tree restricted-role** command to restrict a port from becoming a root port. The default is to allow the DCB interface to become a root port. This procedure affects MSTP only.

To restrict the DCB interface from becoming a root port, run the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree restricted-role** command to restrict the DCB interface from becoming a root port.

```
switch(conf-if-te-1/0/1)# spanning-tree restricted-role
```

Restricting the topology change notification (DCB)

From the DCB interface, use the **spanning-tree restricted-tcn** command to restrict the Topology Change Notification (TCN) BPDUs sent on the interface. By default, the restriction is disabled. This procedure affects MSTP only.

To restrict the TCN BPDUs sent on the DCB interface, perform the following steps in privileged EXEC mode.

1. Enter the **config** command to access global configuration mode.
2. Enter the **interface** command to specify the DCB interface type and slot/port number.

```
switch(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.

```
switch(conf-if-te-1/0/1)# no shutdown
```

4. Enter the **spanning-tree restricted-tcn** command to restrict the TCN BPDUs sent on the DCB interface.

```
switch(conf-if-te-1/0/1)# spanning-tree restricted-tcn
```

Configuring DiST

By default, spanning tree is disabled at the global and interface configuration levels. With respect to Distributed STP (DiST), server ports must be configured as an xSTP edge port.

> NOTE
> DiST is supported on the VCS edge ports only. DiST cannot be enabled on ISL ports participating in the TRILL-based fabric within VCS. DiST does not update the port state of ISL ports.

xSTP can be enabled on the VCS by means of the **protocol spanning-tree** command. An interface begins participating in the spanning tree once it is configured by the **spanning-tree enable** command. Refer to Configuring and managing STP and STP variants on page 151.

The following table describes the behavior of interface based on global and interface level configuration.

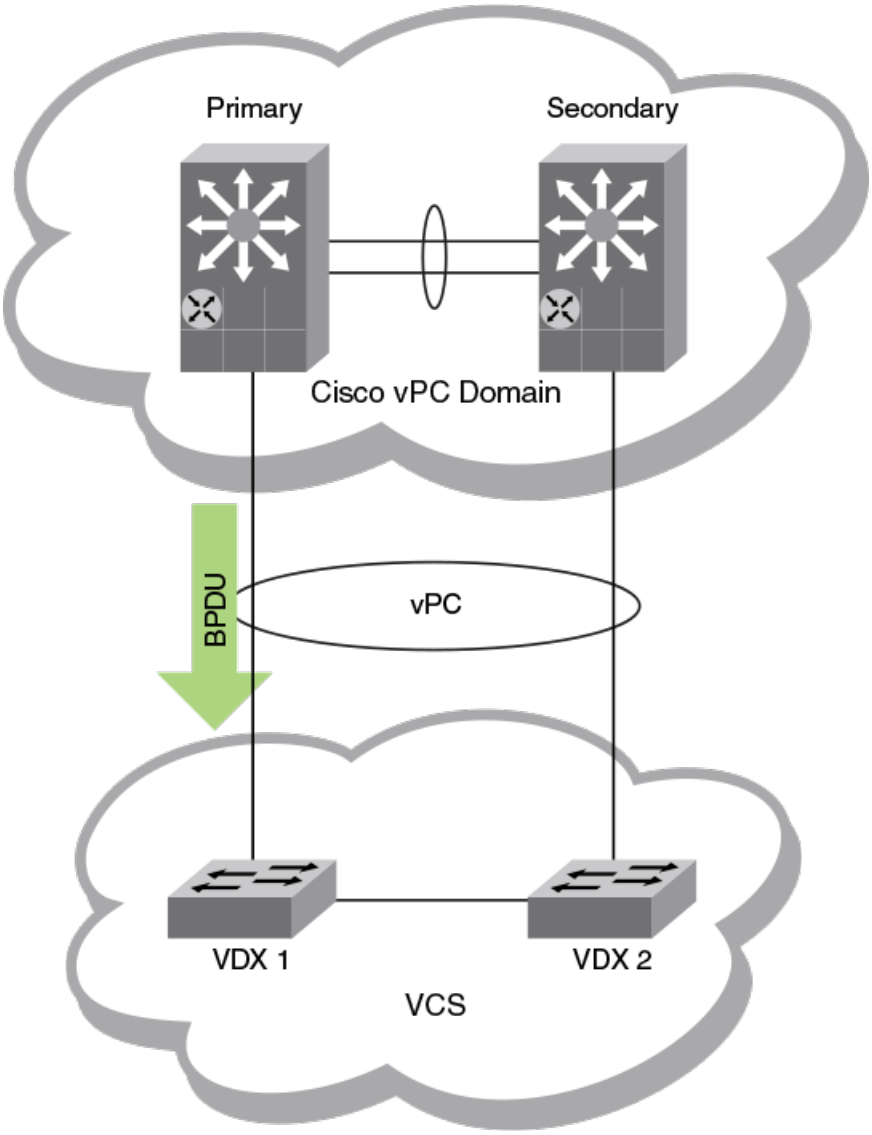
TABLE 21 Interface behavior by global and interface configuration

Global config	Interface config	Interface type where STP BPDU is received	Action
Disable	N/A	Layer 2 (switchport, FCoE)	Flood to all Layer 2 ports
Disable	N/A	Layer 3	Drop
Enable	Disable	Layer 2 (switchport, FCoE)	Drop
Enable	N/A	Layer 3	Drop
Enable	Enable	(switchport, FCoE)	Trap

Cisco Peer-Switch support

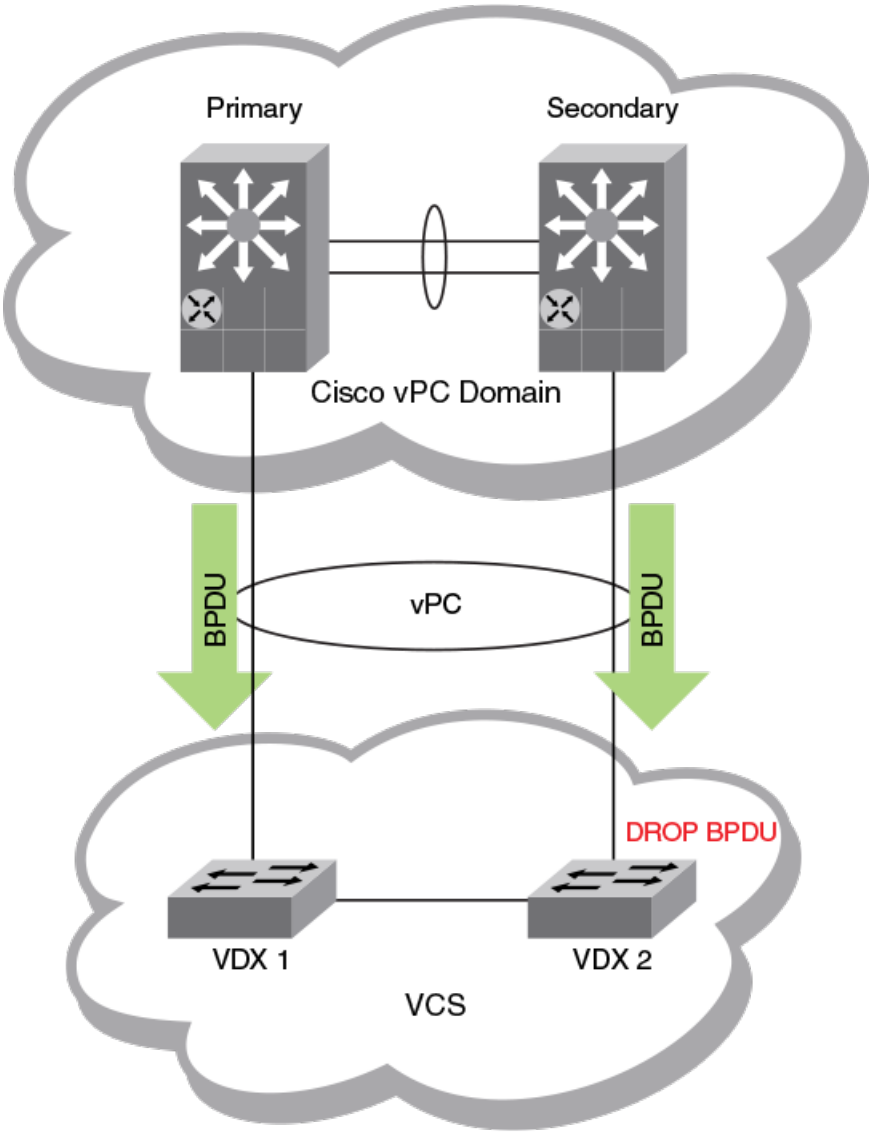
When the Peer-Switch feature is enabled on a Cisco vPC domain, it broadcasts the same BPDUs from both vPC primary and secondary nodes to peer devices. However, a VCS on a VLAG assumes that any logical interface receives only one BPDU from any of its member ports. Consequently, when the VCS receives the two BPDUs from a Cisco vPC domain, it creates a churn of VLAG mastership that increases the CPU load on a Brocade VDX. To avoid this problem, BDPUs received on the VLAG non-master are dropped when the Peer-Switch feature is enabled on the VLAG. The following figure illustrates STP BPDU processing without the Peer-Switch feature.

FIGURE 29 STP BPDU processing without Peer-Switch



In the following figure, where the Peer-Switch feature is enabled, the Brocade VDX 1 receives the BPDU, so it becomes the VLAG master and VDX 2 is set in the non-master state. VDX 1 remains the VLAG master as long as it receives BPDUs.

FIGURE 30 STP BPDU Processing with Peer-Switch



When the Peer-Switch functionality is enabled and the the VLAG Master is selected, BPDUs received on VLAG non-master are dropped unless there is a change in the status of the VLAG master. By default, the Peer-Switch feature functionality is inactive. To activate this function, refer to the **spanning-tree peer-switch** command in the *Network OS Command Reference*.

NOTE

The Peer-Switch feature works only when MSTP is enabled on Cisco switches and a Brocade VDX. It does not work with other flavors of STP. In addition, the Peer-Switch feature is not supported for PVST/RPVST mode unless a Cisco device sends the same BPDU from both the primary and secondary vPC nodes. Currently it sends two different BPDUs. Cisco documentation says that the same BPDU is sent from both primary and secondary nodes of a vPC domain; however, when RPVST is enabled then a message age variable in the BPDU sent from a secondary node is different from that sent from a primary node.

UDLD

UDLD overview

The UniDirectional Link Detection (UDLD) protocol is a nonstandard Layer 2 protocol that detects when a physical link becomes unidirectional by means of the exchange of UDLD protocol data units (PDUs). A unidirectional loop can lead to the creation of a loop in a network, which the Spanning Tree Protocol (STP) could inadvertently allow to occur.

UDLD requirements

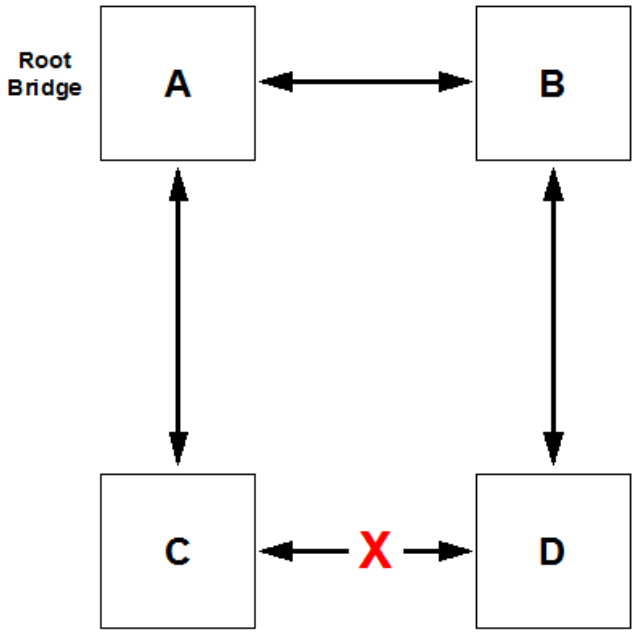
Note the following requirements for UniDirectional Link Detection:

- Network OS 4.0 or later.
- UDLD runs only on physical ports assigned to a port channel.
- UDLD is supported on directly connected switches only.
- In VCS mode, UDLD applies only to edge ports.
- UDLD can interoperate with Brocade IP products.

How UDLD works

The following shows a simple four-switch network in which two paths connect to each switch. STP blocks traffic on as many ports as necessary so that only one operational path exists from the STP root bridge to all nodes in the network.

FIGURE 31 Four-switch example for UDLD



In the figure above, STP detects that the port on switch D that is connected to switch C should be put into a blocked state. Therefore, no data traffic gets transmitted or received on this port. Data traffic remains blocked as long as switch D receives bridge protocol data units (BPDUs) from both switches C and B.

If the link between switch C and switch D becomes unidirectional (for reasons such as hardware failure or incorrect cabling) in the direction from D to C, switch D ages out the status that it was receiving BPDUs from switch C. This eventually causes STP to put the port in a forwarding state, thus allowing all data traffic. This creates a loop for all BUM traffic that enters the network. BUM traffic can go from switch B to switch D to switch C to switch A, and then back to switch B.

To prevent this loop from forming, UDLD can be used to detect that the link between switch C and switch D has become unidirectional.

The UDLD protocol is disabled by default. To use the UDLD protocol, you must first enable the protocol globally and then enable UDLD on each desired individual physical port. For a configuration example, refer to UDLD on page 173.

UDLD determines that a link has become unidirectional if the UDLD daemon stops receiving UDLD PDUs from the other end of the link. The UDLD daemon then blocks the physical link. The physical link remains up but the line protocol goes down. During this time, the link continues to transmit and receive UDLD PDUs.

NOTE
In a VCS environment, the UDLD protocol is applicable only to the edge ports in the VCS. A configuration command to enable the UDLD protocol on a logical port or a non-edge port will be rejected.

Configuring UDLD

Follow the steps below to configure basic UDLD on your switch.

1. Enter global configuration mode by entering the **configure** command from the desired switch:

```
switch# configure
```

2. To enable the UDLD protocol, as well as to enter protocol UDLD configuration mode, enter the **protocol udld** command.

```
switch(config)# protocol udld
```

3. (Optional) You can change the interval at which UDLD PDUs are transmitted from edge ports. The default interval, in counts of one hundred milliseconds, is 5 (500 milliseconds). To change the interval to 2,000 milliseconds, enter the **hello 20** command:

```
switch(config-udld)# hello 20
```

4. You can change the timeout multiplier value to affect the UDLD PDU timeout interval. The UDLD timeout interval is the product of the hello time interval at the other end of the link and the timeout multiplier value. To change the timeout multiplier from the default of 5 to the value 8, run the **multiplier 8** command:

```
switch(config-udld)# multiplier 8
```

5. Enter interface subconfiguration mode for the edge port on which you want to enable UDLD:

```
switch(config-udld)# end
switch# configure
switch(config)# interface te 5/0/1
switch(config-int-te-5/0/1)# udld enable
```

6. Repeat the preceding step for each edge port on which you wish to enable UDLD.

NOTE
When the UDLD protocol is enabled on one end of a link, the timeout period might elapse before the UDLD protocol is enabled on the other end of the link. In this case, the link becomes temporarily blocked. When the UDLD protocol is enabled at the other end of the link and a UDLD PDU is received, UDLD automatically unblocks the link.

Additional UDLD-related commands

Among additional UDLD commands that you can use are the following:

- **clear udld statistics** — Clears either all UDLD statistics or clears the statistics on a specified port.
- **debug udld packet** — Enables debugging for UDLD
- **show debug udld** — Displays UDLD debug status on the switch.
- **show udld** — Displays global UDLD information.
- **show udld interface** — Displays UDLD information for one or all ports.
- **show udld statistics** — Displays either all UDLD statistics or the statistics on a specified port.

For more information about how to use UDLD commands, refer to the *Network OS Command Reference*.

Link Aggregation

Link aggregation overview

Link aggregation allows you to bundle multiple physical Ethernet links to form a single logical trunk providing enhanced performance and redundancy. The aggregated trunk is referred to as a Link Aggregation Group (LAG). The LAG is viewed as a single link by connected devices, the Spanning Tree Protocol, IEEE 802.1Q VLANs, and so on. When one physical link in the LAG fails, the other links stay up and there is no disruption to traffic.

To configure links to form a LAG, the physical links must be the same speed and all links must go to the same neighboring device. Link aggregation can be done by manually configuring the LAG or by dynamically configuring the LAG using the IEEE 802.3ad Link Aggregation Control Protocol (LACP).

When queuing traffic from multiple input sources to the same output port, all input sources are given the same weight, regardless of whether the input source is a single physical link or a trunk with multiple member links.

NOTE
The LAG or LAG interface is also referred to as a *port-channel*.

The benefits of link aggregation are summarized as follows:

- Increased bandwidth (The logical bandwidth can be dynamically changed as the demand changes.)
- Increased availability
- Load sharing
- Rapid configuration and reconfiguration

The Brocade VDX family of switches supports the following trunk types:

- Static, standards-based LAG
- Dynamic, standards-based LAG using LACP
- Static, Brocade-proprietary LAG
- Dynamic, Brocade-proprietary LAG using proprietary enhancements to LACP

Link Aggregation Control Protocol

Link Aggregation Control Protocol (LACP) is an IEEE 802.3ad standards-based protocol that allows two partner systems to dynamically negotiate attributes of physical links between them to form logical trunks. LACP determines whether a link can be aggregated into a LAG. If a link can be aggregated into a LAG, LACP puts the link into the LAG. All links in a LAG inherit the same administrative characteristics.

LACP operates in two modes:

- *Passive mode* — LACP responds to Link Aggregation Control Protocol Data Units (LACPDUs) initiated by its partner system but does not initiate the LACPDU exchange.
- *Active mode* — LACP initiates the LACPDU exchange regardless of whether the partner system sends LACPDUs.

Dynamic link aggregation

Dynamic link aggregation uses LACP to negotiate which links can be added and removed from a LAG. Typically, two partner systems sharing multiple physical Ethernet links can aggregate a number of those physical links using LACP. LACP creates a LAG on both partner systems and identifies the LAG by the LAG ID. All links with the same administrative key and all links that are connected to the same partner switch become members of the LAG. LACP continuously exchanges LACPDUs to monitor the health of each member link.

Static link aggregation

In static link aggregation, links are added into a LAG without exchanging LACPDUs between the partner systems. The distribution and collection of frames on static links is determined by the operational status and administrative state of the link.

LACP configuration guidelines and restrictions

This section applies to standards-based and Brocade-proprietary LAG configurations, except where specifically noted otherwise.

Follow these LACP configuration guidelines and restrictions when configuring LACP:

- All ports on the Brocade VDX hardware can operate only in full-duplex mode.
- On Brocade-proprietary LAGs only, all LAG member links must be part of the same port-group.
- Interfaces configured as "switchport" interfaces cannot be aggregated into a LAG. However, a LAG can be configured as a switchport.

Brocade-proprietary aggregation

Brocade-proprietary aggregation is similar to standards-based link aggregation but differs in how the traffic is distributed. It also has additional rules that member links must meet before they are aggregated:

- The most important rule requires that there is not a significant difference in the length of the fiber between the member links, and that all member links are part of the same port-group.
- A maximum of four Brocade LAGs can be created per port-group.

LAG distribution process and conditions

The LAG aggregator is associated with the collection and distribution of Ethernet frames. The collection and distribution process is required to guarantee the following:

- Inserting and capturing control PDUs.
- Restricting the traffic of a given conversation to a specific link.
- Load balancing between individual links.
- Handling dynamic changes in LAG membership.

On each port, link aggregation control does the following:

- Maintains configuration information to control port aggregation.
- Exchanges configuration information with other devices to form LAGs.
- Attaches ports to and detaches ports from the aggregator when they join or leave a LAG.
- Enables or disables an aggregator's frame collection and distribution functions.

Each link in the Brocade VDX hardware can be associated with a LAG; a link cannot be associated with more than one LAG. The process of adding and removing links to and from a LAG is controlled statically, dynamically, or through LACP.

Each LAG consists of the following components:

- A MAC address that is different from the MAC addresses of the LAG's individual member links.
- An interface index for each link to identify the link to neighboring devices.
- An administrative key for each link. Only links having the same administrative key value can be aggregated into a LAG. On each link configured to use LACP, LACP automatically configures an administrative key value equal to the port-channel identification number.

Virtual LAGs

Configuring a virtual LAG (vLAG) is similar to configuring a LAG. Once the Brocade VCS Fabric detects that the LAG configuration spans multiple switches, the LAG automatically becomes a vLAG.

LACP on the Brocade VCS Fabric emulates a single logical switch by sending the same LACP system ID and sending the same admin and operational key.

Note these vLAG features :

- Only ports with the same speed are aggregated.
- Brocade proprietary LAGs are not available for vLAGs.
- LACP automatically negotiates and forms the vLAG.
- A port-channel interface is created on all the vLAG members.
- The Brocade VCS Fabric relies on you to consistently configure all nodes in the vLAG.
- Similar to static LAGs, vLAGs are not able to detect configuration errors.
- A zero port vLAG is allowed.
- IGMP snooping fits into the primary link of a vLAG to carry multicast traffic.
- Interface statistics are collected and shown per vLAG member switch. The statistics are not aggregated across switches participating in a vLAG.
- In order to provide link and node level redundancy, the Brocade VCS Fabric supports static vLAGs.
- A vLAG can be configured within a virtual fabric. For more information about virtual fabrics, refer to the "Virtual Fabrics" chapter.
- A Brocade VCS Fabric vLAG functions with servers that do not implement LACP because it supports static vLAGs as well.

vLAG scalability

Dynamic virtual LAGs (vLAGs) are made scalable to support a resilient network infrastructure. To achieve scalability of vLAGs, the system must disable the following vLAG operations at the VCS Fabric level:

- vLAG member link-state update
- Primary RBridge selection
- Actor system ID (SID) selection
- Partner system ID (SID) validation

Use the **vlag-commit-mode disable** command to disable the vLAG operations at the VCS Fabric level and to support vLAG scalability. Use the **show running-config vlag-commit-mode** command to view the current vLAG commit-mode state.

The following system error message is generated when multiple partners are detected for a vLAG. This message is displayed when the **vlag-commit-mode disable** command is configured.

```
2015/01/04-16:33:23, [LACP-1001], 783124, SW/0 | Active | DCE, ERROR, sw0, vLAG multiple partner
detected on Po 110
```

If this system error message appears because of a potential multi-partner vLAG conflict, the user should issue the **show vlag-partner-info** command to verify the multi-partner vLAG configuration and resolve any issues, as shown in the following example.

```
device# show vlag-partner-info
Port-channel 110
RBridge 1: Partner System ID - 0xffff,00-05-33-48-71-a8 Key 0009
RBridge 4: Partner System ID - 0xffff,11-22-33-44-55-66 Key 0009
RBridge 5: Partner System ID - 0xffff,00-05-33-48-71-a8 Key 0009
```

NOTE

The **vlag-commit-mode disable** command cannot be configured if the **no vlag ignore-split** command is configured on any port-channel in the VCS Fabric. Similarly, the **no vlag ignore-split** command cannot be configured if the **vlag commit-mode disable** command is configured.

IP over port-channel

Beginning with Network OS 7.0.0 and the introduction of IP Fabrics, support is provided over port-channels (Layer 2) for Layer 3 protocols.

Support for IP over port-channel provides the following advantages:

- Increases bandwidth between two RBridges.
- Provides fault tolerance, so that there is zero loss until the last member of the port-channel remains.
- Provides for dynamic bandwidth, through the removal or addition of port-channel members. (The upper layers do not need to know about the members, as these are device-dependent.)
- Provides load balancing.

This feature provides support for the following:

- Standard and Brocade port-channels, both static and dynamic
- IPv4 and IPv6 addressing
- Configuration and show commands as are currently supported for physical ports
- High availability

Support is provided for the following Layer 3 protocols:

- ARP/ND
- BFD
- BGP
- DHCP
- ICMP
- IGMP
- OSPFv2/v3
- PIM
- VRRP

In addition, support is provided for route-map policy.

Limitations

Note the following limitations:

- IP over vLAGs is not supported. If a port-channel with member interfaces from more than one node in logical chassis cluster mode, IPv4/IPv6 configurations are not allowed. If a port-channel (vLAG) with members from two nodes becomes segmented, it is no longer a vLAG operationally and IPv4/IPv6 configurations are not allowed.
- sFlow is not supported for Layer 3 port-channels.
- Layer 3 configurations, including IPv4/IPv6 address configurations, are not allowed on an "empty" port-channel (that is, one that has not yet been configured).
- A port-channel cannot be unconfigured until all Layer 3 configurations are removed from it.

Supported commands

The following commands, organized largely by protocol, support IP over port-channel.

IP routing commands

- ip route
- ipv6 route

Interface commands

- clear ipv6 counters interface port-channel
- ip address
- ip address secondary
- ip mtu
- ipv6 address
- ipv6 address secondary
- ipv6 address use-link-local-only
- ipv6 address eui-64
- ipv6 address eui-64 secondary
- ipv6 address link-local
- ipv6 address anycast
- show ip interface
- show ipv6 interface
- show ip interface brief
- show ipv6 interface brief
- show ipv6 counters interface port-channel
- show port port-channel
- show port-channel
- show port-channel-redundancy-group

ARP/ND commands

- arp interface port-channel
- ip proxy-arp
- ip arp-aging-timeout
- show arp port-channel
- show ipv6 neighbor port-channel

BGP commands

- neighbor

DHCP commands

- ip dhcp relay

ICMP commands

- ip icmp

IGMP commands

- ip igmp immediate-leave
- ip igmp last-member-query-count
- ip igmp last-member-query-interval
- ip igmp query-interval
- ip igmp query-max-response-time
- ip igmp robustness-variable
- ip igmp startup-query-count
- ip igmp startup-query-interval
- ip igmp static-group
- show ip igmp interface port-channel
- show ip igmp groups interface port-channel

OSPFv2 commands

- ip ospf active
- ip ospf area
- ip ospf auth-change-wait-time
- ip ospf autentication-key
- ip ospf bfd
- ip ospf cost
- ip ospf database-filter
- ip ospf dead-interval
- ip ospf hello-interval
- ip ospf md5-authentication
- ip ospf mtu-ignore
- ip ospf network
- ip ospf passive
- ip ospf priority
- ip ospf retransmit-interval
- ip ospf transmit-delay
- clear ip ospf counters port-channel
- show ip ospf interface port-channel
- show ip ospf neighbor port-channel
- show debug ip ospf internal interface port-channel

OSPFv3 commands

- ipv6 ospf active
- ipv6 ospf area
- ipv6 ospf autentication
- ipv6 ospf bfd
- ipv6 ospf cost
- ipv6 ospf dead-interval
- ipv6 ospf hello-interval
- ipv6 ospf instance
- ipv6 ospf mtu-ignore
- ipv6 ospf network
- ipv6 ospf passive
- ipv6 ospf priority
- ipv6 ospf retransmit-interval
- ipv6 ospf suppress-linklsa
- ipv6 ospf transmit-delay
- clear ipv6 ospf counts neighbor interface port-channel
- clear ipv6 ospf neighbor interface port-channel
- show ipv6 ospf interface port-channel
- show ipv6 ospf neighbor interface port-channel

PIM commands

- ip multicast-booundary
- ip pim-sparse
- ip pim dr-priority
- ip pim neighbor-filter
- show ip pim-sparse interface port-channel
- show ip pim neighbor interface port-channel

Route-map policy commands

- ip policy route-map
- ipv6 policy route-map
- match interface port-channel
- show route-map interface port-channel

VRRP commands

- clear vrrp statistics interface port-channel
- clear ipv6 vrrp statistics interface port-channel
- debug vrrp packets interface port-channel
- debug ipv6 vrrp packets interface port-channel
- ipv6 vrrp-group
- show ipv6 vrrp interface port-channel

- **show vrrp interface port-channel**
- **vrrp-group**

Ethernet Segment Identifiers (ESIs) for BGP routing

ESIs are used to identify the links connecting multiple ToR devices to a server in a BGP EVPN environment.

If a server is connected to more than one ToR devices over a vLAG interface, the set of links that attaches the server to these devices is referred to as an Ethernet Segment (ES), which is associated with a globally unique Ethernet Segment Identifier, or ESI. When BGP routing is used to support Ethernet Virtual Private Network (EVPN) deployments, each BGP router advertises an Ethernet Segment Route (ESR) to inform other devices that it is connected to that ES. This allows different BGP routers to discover whether they are connected to the same ES. The default ESI value is 0 (Single Home). (The links that connect single-homed devices to server are not required to have an ESI value associated with them.)

The ESI value an be configured manually by means of the **esi** command in port-channel configuration mode. For details, see "Configuring an ESI on a port-channel for BGP routing" later in this chapter.

> NOTE
> The ESI value can also be derived automatically by means of the LACP Partner SystemID/Port Key. Because this value is the same when the host is dual-homed, the two peer BGP routers derive identical ESI values for the ES and autodiscovery is enabled.

Link aggregation setup

The following sections discuss how to set up link aggregation.

vLAG configuration overview

Network OS 4.0 and later supports the option of setting the "Allowed Speed" of the port-channel to either 1 Gbps or 10 Gbps. The default is 10 Gbps. If the port-channel is 1 Gbps, then the speed needs to be configured before the port-channel is enabled. Otherwise, the physical links are throttled down because of a speed mismatch. Refer to the *Network OS Command Reference* for information on the **speed** command.

The following conditions and requirements should be kept in mind when configuring vLAGs:

- FCoE and DCB capabilities are not supported by vLAG. FCoE traffic is treated similarly to normal LAN data traffic.
- Static vLAGs are not supported on internal ports.
- Perform this procedure on all member nodes of a vLAG.

Configuring vLAGs

To configure a vLAG, perform the following steps:

1. Change to global configuration mode.
2. Configure a LAG between two switches within the Brocade VCS Fabric.

   Refer to LAG distribution process and conditions on page 178 for more information. Once the Brocade VCS Fabric detects that the LAG configuration spans multiple switches, the LAG automatically becomes a vLAG.
3. Enter **interface port-channel** *ID* on every switch in the vLAG to configure them to treat FCoE MAC addresses as being multi-homed hosts, similar to LAN traffic.

The default configuration is to treat FCoE traffic as non-vLAG traffic.

```
switch(config)# interface port-channel 10
```

4. Enter **end** to return to privileged EXEC mode.

```
switch(config-Port-channel-10)# end
switch#
```

5. Enter the **show port-channel** *detail* command to verify the port-channel details.

```
switch# show port-channel detail
LACP Aggregator: Po 27
 Aggregator type: Standard
 Ignore-split is disabled
 Actor System ID - 0x8000,00-05-33-6f-18-18
 Admin Key: 0027 - Oper Key 0027
 Receive link count: 4 - Transmit link count: 4
 Individual: 0 - Ready: 1
 Partner System ID - 0x8000,00-05-1e-cd-6e-9f
 Partner Oper Key 0027
 Member ports on rbridge-id 231:
  Link: Te 231/0/22 (0xE718160201) sync: 1  *
  Link: Te 231/0/23 (0xE718170202) sync: 1
  Link: Te 231/0/36 (0xE718240305) sync: 1
  Link: Te 231/0/37 (0xE718250306) sync: 1
```

6. Enter the **show port port-channel** command to verify the port-channel interface details.

```
switch# show port port-channel tengigabitethernet 1/0/21
LACP link info: te0/21 -0x18150014
Actor System ID: 0x8000,01-e0-52-00-01-00
Actor System ID Mapped Id: 0
Partner System ID: 0x0001,01-80-c2-00-00-01
Actor priority: 0x8000 (32768)
Admin key: 0x000a (10) Operkey: 0x0000 (0)
Receive machine state : Current
Periodic Transmission machine state : Slow periodic
Muxmachine state : Collecting/Distr
Admin state: ACT:1 TIM:0 AGG:1 SYN:0 COL:0 DIS:0 DEF:1 EXP:0
Operstate: ACT:1 TIM:0 AGG:1 SYN:1 COL:1 DIS:1 DEF:0 EXP:0
Partner operstate: ACT:1 TIM:0 AGG:1 SYN:1 COL:1 DIS:1 DEF:0 EXP:0
Partner oper port: 100
```

Configuring vLAGs to minimize packet loss

This topic provides background on configuring a vLAG to minimize packet loss.

In scenarios where a vLAG spans more than one node, the **vlag ignore-split** command minimizes the extent of packet loss in the event of one of the nodes in the vLAG going down, and also reduces vLAG failover downtime. The scope of this configuration is per port-channel on LACP-based vLAGS.

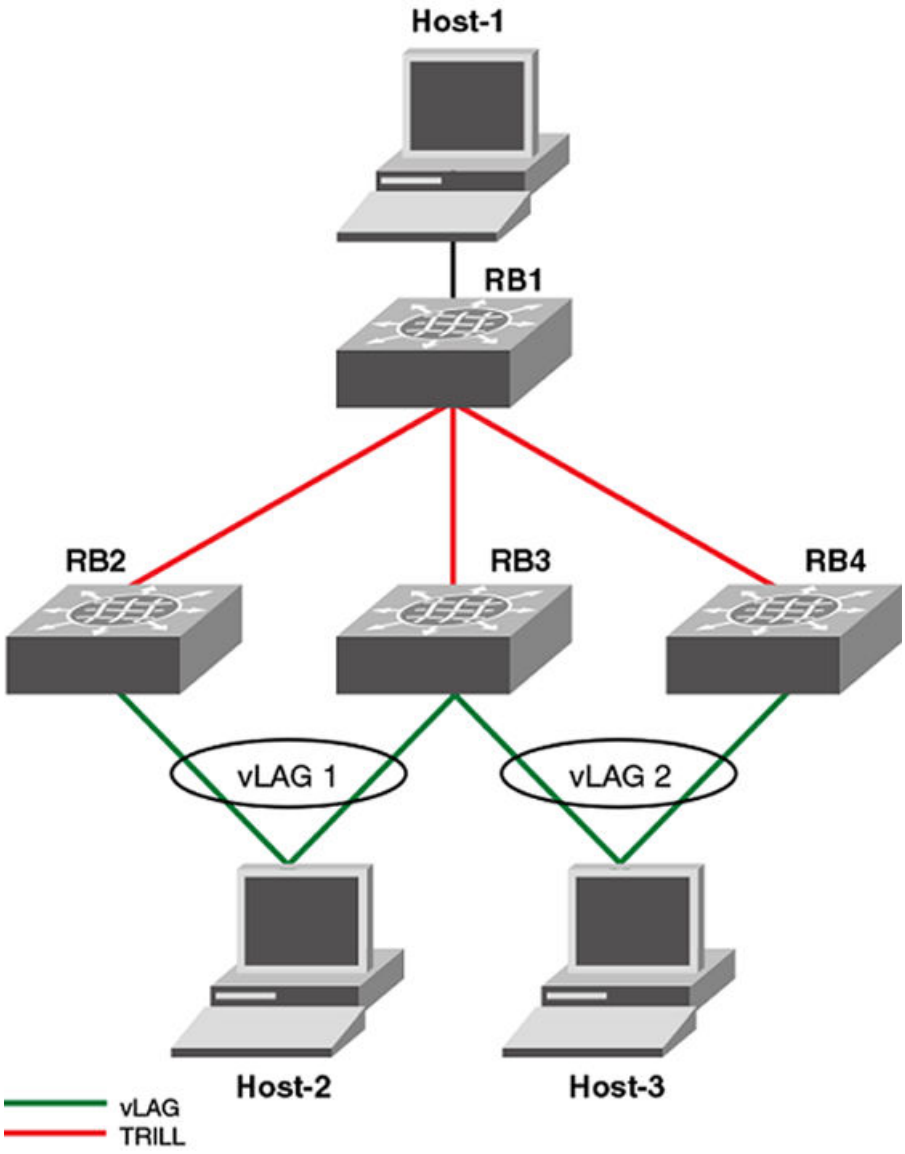
In a case where connectivity between nodes is lost because of a fabric split (as opposed to one of members going down), there will be duplication of multicast/broadcast packets. Brocade recommends that you build redundancy in the fabric so that individual links are not single points of failure.

NOTE

With **ignore-split** active, a vLAG node reboot can result in a more than one-second loss while interoperating with a Linux server/nic-team/CNA, because of premature egress of traffic from the server.

The figure below displays a dual-vLAG configuration with three legs of RB2, RB3, and RB4. If RB2, RB3, or RB4 reboots while Host-1 is communicating to Host-2 or Host3, a momentary traffic disruption may occur.

FIGURE 32 vLAG configuration of the ignore-split feature



To reduce vLAG failover down time, you must configure **ignore-split** on all of the legs in the vLAG (RB2, RB3 and RB4 in this case).

NOTE
By default, **vlag ignore-split** is already activated in VCS.

Configuring the vLAG ignore-split feature on page 186 walks you through setting up the vLAG ignore-split feature.

Configuring the vLAG ignore-split feature

This topic describes how to configure the vLAG ignore-split feature.

The switch must be in global configuration mode.

To configure the vLAG ignore-split feature, perform the following steps.

NOTE
The following example is based on the illustration in Configuring vLAGs on page 184.

1. Log in to RB2, the first leg of the vLAG 1.
2. Access the port-channel for the first leg.

```
switch(config)# interface port-channel 1
```

3. Activate vLAG ignore split.

```
switch(config-Port-channel-1)# vlag ignore-split
```

4. Log in to RB3, the second leg of vLAG 1.
5. Access the port-channel for the second leg.

```
switch(config)# interface port-channel 2
```

6. Activate vLAG ignore split.

```
switch(config-Port-channel-2)# vlag ignore-split
```

7. Access the port-channel for the third leg.

```
switch(config)# interface port-channel 3
```

8. Activate vLAG ignore split.

```
switch(config-Port-channel-3)# vlag ignore-split
```

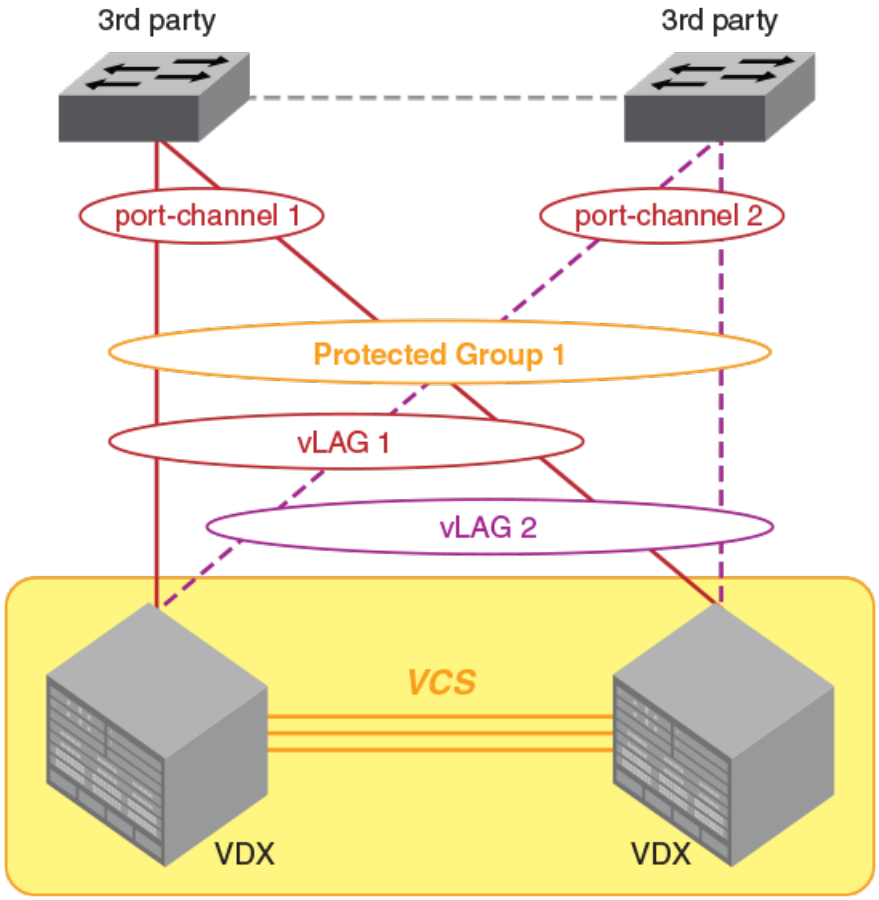
Port-channel redundancy groups

A port-channel redundancy group is used as an alternative to the spanning-tree-protocol for faster convergence and failover in the critical path of the redundant network.

Port-channel redundancy groups minimize disruption to the network by protecting critical links from data loss.

A port-channel redundancy group is a pair vLAGs (or LAGs) configured to act as one active and one backup vLAG connection. The vLAGs and the corresponding port-channels must be configured before the port-channel redundancy groups can be configured. At any point in time only one vLAG is actively forwarding the traffic and the other vLAG is in a discarding state. If the active vLAG goes down, the backup vLAG take over, by changing the vLAG port state from discarding to forwarding. The failover occurs within milliseconds. The port-channel redundancy group controls the port state of the member vLAGs to avoid looping errors.

FIGURE 33 Port-channel redundancy group configuration



There is no restriction on the membership of the active or backup vLAG in the fabric. Either the active or the backup vLAG can coexist in a single RBridge, or they can exist in different RBridges.

Consider the following guidelines and restrictions when configuring a port-channel redundancy group:

- The maximum number of supported port-channel redundancy groups is 255.
- Port-channel redundancy groups are supported only on port-channel interfaces.
- If you designate the active vLAG when configuring the port-channel redundancy group, it will always resume the active vLAG role after a failure, and the other vLAG will return to the backup role.
- If you do not designate one of the vLAGs as "active" then the system assigns the vLAG that comes first in the protected group as the active vLAG, and the second becomes the backup vLAG. If the active vLAG fails and the backup vLAG takes over, the backup vLAG will retain the active role when the original vLAG resets.
- Brocade trunks must not be a member of a protected group.
- Membership in a port-channel redundancy group cannot be altered when the group is in the active state. To modify the protected group or to change the "active" role, first deactivate the protected group.
- Port-channel redundancy groups do not support the STP family of protocols.

Configuring port-channel redundancy groups

Two port-channels are placed into a protected group. In this protected group, one port-channel acts as the active link, and the other port-channel acst as the standby link.

This task assumes that the port channels have already been created using the instructions in Configuring vLAGs on page 184.

To configure a port-channel redundancy group, perform the following task in global configuration mode.

1. Open the port-channel redundancy-group configuration mode with the **port-channel-redundancy-group** command.

```
switch(config)#port-channel-redundancy-group 27
switch(config-port-channel-redundancy-group-27)#
```

2. Add the active port-channel to the group with the **port-channel** command.

```
switch(config-port-channel-redundancy-group-27)# port-channel 3 active
```

3. Add the standby port-channel to the group with the **port-channel** command.

```
switch(config-port-channel-redundancy-group-27)# port-channel 5
```

4. Activate the port-channel redundancy group with the **activate** command.

```
switch(config-port-channel-redundancy-group-27)# activate
```

5. Confirm the configuration with the **show port-channel-redundancy-group** command.

```
switch#show port-channel-redundancy-group 27
Group ID                      : 27
Member Ports                  : Port-channel 3, Port-channel 5
Configured Active Port-channel: Port-channel 3
Current Active Port-channel   : Port-channel 3
Backup Port-channel           : Port-channel 5
```

Configuring an ESI on a port-channel for BGP routing

Ethernet Segment Identifiers (ESIs) are used to identify the links connecting multiple ToR devices to a server in a multihomed BGP EVPN environment. Use this task to configure ESI values manually on a port-channel interface if LACP autodiscovery is not deployed.

For details, see "Ethernet Segment Identifiers for BGP routing" earlier in this chapter.

1. From global configuration mode, specify a port-channel interface and enter port-channel configuration mode.

```
device# config terminal
device(config)# interface port-channel 1
device(config-Port-channel-1)#
```

2. Enter the **esi** command and specify an appropriate hexadecimal ESI value.

```
device(config-Port-channel-1)# esi 00:11:22:33:44:55:66:77:88:99
```

3. Repeat the above as appropriate on all port-channel interfaces to be configured manually.
4. Do any of the following to manage the configuration.
   a) Use the **esi auto lacp** command to enable automatic ESI assignment.

```
device(config-Port-channel-1)# esi auto lacp
```

   b) Use the **no esi** *<value>* command to remove the ESI value from the interface and allow a new number to be assigned.

```
device(config-Port-channel-1)# no esi 00:11:22:33:44:55:66:77:88:99
```

c) Use the **no esi auto lacp** command to reenable the assignment of the most current ESI to the interface.

```
device(config-Port-channel-1)# no esi auto lacp
```

Configuring load balancing on a remote RBridge

This feature allows you to configure the load-balancing feature on a remote RBridge, which is not a member of the vLAG (also known as a non-local RBridge), to forward traffic to a vLAG. To distribute the traffic among the possible paths towards the VLAG, you can configure the vLAG load-balancing flavor on RB2. Available flavors are listed below.

TABLE 22 Load balancing flavors

Flavor	Definition
dst-mac-vid	Destination MAC address and VID-based load balancing.
src-mac-vid	Source MAC address and VID-based load balancing.
src-dst-mac-vid	Source and Destination MAC address and VID-based load balancing.
src-dst-ip	Source and Destination IP address-based load balancing.
src-dst-ip-mac-vid	Source and Destination IP and MAC address and VID-based load balancing.
src-dst-ip-port	Source and Destination IP and TCP/UDP port-based load balancing.
src-dst-ip-mac-vid-port	Source and Destination IP, MAC address, VID and TCP/UDP port-based load balancing.

Additionally, an RBridge can be set to a different flavor for different vLAGs present in the cluster. This feature is available for each RBridge and each VLAG, so different load-balancing flavors can be set for traffic directed towards different VLAGs. The **show running-config rbridge-id** *rbridgeID* command displays the configuration information.

NOTE

When configuring load balancing on a Brocade VDX 6740, it should be configured consistently for all port-channels on the switch. These switches support one load-balancing scheme at a time, and apply the last loaded load-balancing scheme to all port-channels on the switch. This is not required for the Brocade VDX 8770 platform, as it supports multiple port-channel load-balancing schemes.

The following example sets the flavor to "destination MAC address and VID-based load balancing."

```
switch(config)# rbridge-id 2
switch(config-rbridge-id-2)# fabric port-channel 20 load-balance dst-mac-vid
switch(config-rbridge-id-2)# end
switch# show running-config rbridge-id 2
rbridge-id 2
 interface-nodespecific ns-vlan 10
 interface-nodespecific ns-ethernet 100
 fabric vlag 10 load-balance src-dst-mac-vid
 fabric vlag 20 load-balance dst-mac-vid
 no protocol vrrp
switch# show fabric port-channel load-balance 10
Fabric Vlag Load-Balance Information
------------------------------
Rbridge-Id        :  2
Vlag              : 10
Load-Balance Flavor : Source and Destination MAC address and VID based load balancing

switch# show fabric port-channel all

Fabric Vlag Load-Balance Information
------------------------------
Rbridge-Id         :  2
Vlag               : 10
```

Configuring and managing LACP

The following sections discuss working with the Link Aggregation Control Protocol (LACP) on Brocade devices.

Understanding the default LACP configuration

The table below lists the default LACP configuration. Consider this when making changes to the defaults.

TABLE 23 Default LACP configuration

Parameter	Default setting
System priority	32768
Port priority	32768
Timeout	Long (standard LAG) or short (Brocade LAG)

Enabling LACP on a DCB interface

The switch must be in privileged EXEC mode.

To add additional interfaces to an existing LAG, repeat this procedure using the same LAG group number for the new interfaces.

Enter the **copy running-config startup-config** command to save your configuration.

To enable LACP on a DCB interface, perform the following steps.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command, specifying the DCB interface type and RBridge/slot/port.

```
switch(config)# interface tengigabitethernet 5/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.
4. Enter the **channel-group** command to configure the LACP for the DCB interface.

```
switch(conf-if-te-5/0/1)# channel-group 4 mode active type standard
```

Configuring neighbor discovery for LACP on an interface

You may need to disable neighbor discovery for Brocade devices on a per-interface basis so that the Brocade VDX does not to bring up its ports in an uncontrolled fashion until the fabric completely forms. This option is needed when an unconditional EtherChannel is configured between the VCS fabric and an end node, usually ESX or Hypervisors, which does not support LACP. If a Brocade VDX brings up its ports unexpectedly, the data traffic may be compromised.

1. Enter global configuration mode.

```
device# configure terminal
```

2. Enter interface subtype configuration mode for the port.

```
device(config)# interface tengigabitethernet 1/0/18
```

3. Use the **fabric neighbor-discovery disable** command to disable neighbor discovery.

```
device(config-if-te-1/0/18)# fabric neighbor-discovery disable
```

4. Once the fabric has come online, use the **no fabric neighbor-discovery disable** command to reenable neighbor discovery on the interface.

```
device(config-if-te-1/0/18)# no fabric neighbor-discovery disable
```

Configuring the LACP system priority

The switch must be in privileged EXEC mode.

You configure the LACP system priority on each switch running LACP. LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other switches.

The system priority value must be a number in the range of 1 through 65535. The higher the number, the lower the priority. The default priority is 32768.

To configure the global LACP system priority, perform the following steps:

1. Enter the **configure terminal** command to access global configuration mode.
2. Specify the LACP system priority.

```
switch(config)# lacp system-priority 25000
```

Configuring the LACP timeout period on a DCB interface

The switch must be in privileged EXEC mode.

The LACP timeout period indicates how long LACP waits before timing out the neighboring device. The **short** timeout period is 3 seconds and the **long** timeout period is 90 seconds. The default is **long**.

To configure the LACP timeout period on a Data Center Bridging (DCB) interface, perform the following steps:

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **interface** command, specifying the DCB interface type and RBridge/slot/port.

```
switch(config)# interface tengigabitethernet 5/0/1
```

3. Enter the **no shutdown** command to enable the DCB interface.
4. Specify the LACP timeout period for the DCB interface.

```
switch(conf-if-te-5/0/1)# lacp timeout short
```

Clearing LACP counter statistics on a LAG

This topic describes how to clear LACP counter statistics on a single LAG.

To clear LACP counter statistics on a LAG, use the following command:

Enter the **clear lacp** *LAG_group_number* **counters** command to clear the LACP counter statistics for the specified LAG group number.

```
switch# clear lacp 42 counters
```

Clearing LACP counter statistics on all LAG groups

This topic describes how to clear the LACP counter statistics for all LAG groups.

To clear LACP counter statistics on all LAG groups, use the following command:

Enter the **clear lacp counter** command to clear the LACP counter statistics for all LAG groups.

```
switch# clear lacp counter
```

Displaying LACP information

You can use the **show** command in privileged EXEC mode to display Link Aggregation Control Protocol (LACP) information.

- Enter the **show lacp sys-id** command to display the LACP system ID and priority.

```
switch# show lacp sys-id
% System 8000,00-05-1e-76-1a-a6
```

- Enter the **show lacp counter** command to display the LACP system ID and priority.

```
switch# show lacp counter
 Traffic Statistics
Port           LACPDUs               Marker               Pckt err
            Sent     Recv          Sent     Recv          Sent     Recv
12          123      0             2        0             0        0
```

Refer to the *Network OS Command Reference* for more information.

Troubleshooting LACP

To troubleshoot problems with your LACP configuration, use the following troubleshooting tips.

If a standard IEEE 802.3ad-based dynamic trunk is configured on a link and the link is not able to join the LAG, do the following:

- Make sure that both ends of the link are configured as **standard** for the trunk type.
- Make sure that both ends of the link are *not* configured for **passive** mode. They must be configured as **active** /**active**, **active** / **passive**, or **passive** /**active**.
- Make sure that the port-channel interface is in the administrative "up" state by ensuring that the **no shutdown** command was entered on the interface on both ends of the link.
- Make sure the speed parameter is configured to 1000 if the port-channel is using the gigabit interface.
- Make sure that the links that are part of the LAG are connected to the same neighboring switch.
- Make sure that the system ID of the switches connected by the link is unique. You can verify this by entering the **show lacp sys-id** command on both switches.
- You can verify the system ID of the switches in the Brocade VCS Fabric cluster with the **show lacp sys-id** command.
- Make sure that LACPDUs are being received and transmitted on both ends of the link and that there are no error PDUs. You can verify this by entering the **show lacp counters** *number* command and looking at the receive mode (rx) and transmit mode (tx) statistics. The statistics should be incrementing and should not be at zero or a fixed value. If the PDU rx count is not incrementing, check the interface for possible CRC errors by entering the **show interface** *link-name* command on the neighboring switch. If the PDU tx count is not incrementing, check the operational status of the link by entering the **show interface** *link-name* command and verifying that the interface status is "up."

If a Brocade-based dynamic trunk is configured on a link and the link is not able to join the LAG, do the following:

- Make sure that both ends of the link are configured as **Brocade** for trunk type.

- Make sure that both ends of the link are *not* configured for **passive** mode. They must be configured as **active** /**active**, **active** / **passive**, or **passive** /**active**.
- Make sure that the port-channel interface is in the administrative "up" state by ensuring that the **no shutdown** command was entered on the interface on both ends of the link.
- Make sure that the links that are part of the LAG are connected to the same neighboring switch.
- Make sure that the system ID of the switches connected by the link is unique. This can be verified by entering the **show lacp sys-id** command on both switches.
- Make sure that LACPDUs are being received and transmitted on both ends of the link and there are no error PDUs. This can be verified by entering the **show lacp counters** *number* command and looking at the rx and tx statistics. The statistics should be incrementing and should not be at zero or a fixed value. If the PDU rx count is not incrementing, check the interface for possible CRC errors by entering the **show interface** *link-name* command on the neighboring switch.
- Make sure that the fiber length of the link has a deskew value of 7 microseconds. If it does not, the link will not be able to join the LAG and the following RASlog message is generated: `Deskew calculation failed for link <link-name>`.

When a link has this problem, the **show port-channel** command displays the following message:

```
Mux machine state: Deskew not OK.
```

If a Brocade-based static trunk is configured on a link and the link is not able to join the LAG, do the following:

- Make sure that both ends of the link are configured as **Brocade** for trunk type and verify that the mode is "on."
- Make sure that the port-channel interface is in the administrative "up" state by ensuring that the **no shutdown** command was entered on the interface on both ends of the link.

If a standards-based static trunk is configured on a link and the link is not able to join the LAG, do the following:

- Make sure that both ends of the link are configured as **standard** for trunk type and verify that the mode is "on."
- Make sure that the port-channel interface is in the administrative "up" state by ensuring that the **no shutdown** command was entered on the interface on both ends of the link.

LLDP

LLDP overview

The IEEE 802.1AB Link Layer Discovery Protocol (LLDP) enhances the ability of network management tools to discover and maintain accurate network topologies and simplify LAN troubleshooting in multi-vendor environments. To efficiently and effectively operate the various devices in a LAN you must ensure the correct and valid configuration of the protocols and applications that are enabled on these devices. With Layer 2 networks expanding dramatically, it is difficult for a network administrator to statically monitor and configure each device in the network.

Using LLDP, network devices such as routers and switches advertise information about themselves to other network devices and store the information they discover. Details such as device configuration, device capabilities, and device identification are advertised. LLDP defines the following:

- A common set of advertisement messages.
- A protocol for transmitting the advertisements.
- A method for storing the information contained in received advertisements.

NOTE
LLDP runs over the data-link layer which allows two devices running different network layer protocols to learn about each other.

LLDP information is transmitted periodically and stored for a finite period. Every time a device receives an LLDP advertisement frame, it stores the information and initializes a timer. If the timer reaches the time to live (TTL) value, the LLDP device deletes the stored information ensuring that only valid and current LLDP information is stored in network devices and is available to network management systems.

Layer 2 topology mapping

The LLDP protocol lets network management systems accurately discover and model Layer 2 network topologies. As LLDP devices transmit and receive advertisements, the devices store information they discover about their neighbors. Advertisement data such as a neighbor's management address, device type, and port identification is useful in determining what neighboring devices are in the network.

NOTE
The Brocade LLDP implementation supports up to two neighbors.

The higher level management tools, such as the Brocade Network Advisor, can query the LLDP information to draw Layer 2 physical topologies. The management tools can continue to query a neighboring device through the device's management address provided in the LLDP information exchange. As this process is repeated, the complete Layer 2 topology is mapped.

In LLDP the link discovery is achieved through the exchange of link-level information between two link partners. The link-level information is refreshed periodically to reflect any dynamic changes in link-level parameters. The basic format for exchanging information in LLDP is in the form of a type, length, value (TLV) field.

LLDP keeps a database for both local and remote configurations. The LLDP standard currently supports three categories of TLVs. Brocade's LLDP implementation adds a proprietary Brocade extension TLV set. The four TLV sets are described as follows:

- Basic management TLV set — This set provides information to map the Layer 2 topology and includes the following TLVs:

  - Chassis ID TLV — Provides the ID for the switch or router where the port resides. This is a mandatory TLV.
  - Port description TLV — Provides a description of the port in an alphanumeric format. If the LAN device supports RFC-2863, the port description TLV value equals the "ifDescr" object. This is a mandatory TLV.
  - System name TLV — Provides the system-assigned name in an alphanumeric format. If the LAN device supports RFC-3418, the system name TLV value equals the "sysName" object. This is an optional TLV.
  - System description TLV — Provides a description of the network entity in an alphanumeric format. This includes system name, hardware version, operating system, and supported networking software. If the LAN device supports RFC-3418, the value equals the "sysDescr" object. This is an optional TLV.
  - System capabilities TLV — Indicates the primary functions of the device and whether these functions are enabled in the device. The capabilities are indicated by two octets. The first octet indicates Other, Repeater, Bridge, WLAN AP, Router, Telephone, DOCSIS cable device, and Station, respectively. The second octet is reserved. This is an optional TLV.
  - Management address TLV — Indicates the addresses of the local switch. Remote switches can use this address to obtain information related to the local switch. This is an optional TLV.
- IEEE 802.1 organizational TLV set — This set provides information to detect mismatched settings between local and remote devices. A trap or event can be reported once a mismatch is detected. This is an optional TLV. This set includes the following TLVs:
  - Port VLANID TLV — Indicates the port VLAN ID (PVID) that is associated with an untagged or priority tagged data frame received on the VLAN port.
  - PPVLAN ID TLV — Indicates the port- and protocol-based VLAN ID (PPVID) that is associated with an untagged or priority tagged data frame received on the VLAN port. The TLV supports a "flags" field that indicates whether the port is capable of supporting port- and protocol-based VLANs (PPVLANs) and whether one or more PPVLANs are enabled. The number of PPVLAN ID TLVs in a Link Layer Discovery Protocol Data Unit (LLDPDU) corresponds to the number of the PPVLANs enabled on the port.
  - VLAN name TLV — Indicates the assigned name of any VLAN on the device. If the LAN device supports RFC-2674, the value equals the "dot1QVLANStaticName" object. The number of VLAN name TLVs in an LLDPDU corresponds to the number of VLANs enabled on the port.
  - Protocol identity TLV — Indicates the set of protocols that are accessible at the device's port. The protocol identity field in the TLV contains a number of octets after the Layer 2 address that can enable the receiving device to recognize the protocol. For example, a device that wishes to advertise the spanning tree protocol includes at least eight octets: 802.3 length (two octets), LLC addresses (two octets), 802.3 control (one octet), protocol ID (two octets), and the protocol version (one octet).
- IEEE 802.3 organizational TLV set — This is an optional TLV set. This set includes the following TLVs:
  - MAC/PHY configuration/status TLV — Indicates duplex and bit rate capabilities and the current duplex and bit rate settings of the local interface. It also indicates whether the current settings were configured through auto-negotiation or through manual configuration.
  - Power through media dependent interface (MDI) TLV — Indicates the power capabilities of the LAN device.
  - Link aggregation TLV — Indicates whether the link (associated with the port on which the LLDPDU is transmitted) can be aggregated. It also indicates whether the link is currently aggregated and provides the aggregated port identifier if the link is aggregated.
  - Maximum Ethernet frame size TLV — Indicates the maximum frame size capability of the device's MAC and PHY implementation.

DCBX

Storage traffic requires a lossless communication which is provided by DCB. The Data Center Bridging (DCB) Capability Exchange Protocol (DCBX) is used to exchange DCB-related parameters with neighbors to achieve more efficient scheduling and a priority-based flow control for link traffic.

DCBX uses LLDP to exchange parameters between two link peers; DCBX is built on the LLDP infrastructure for the exchange of information. DCBX-exchanged parameters are packaged into organizationally specific TLVs. The DCBX protocol requires an acknowledgment from the other side of the link, therefore LLDP is turned on in both transmit and receive directions. DCBX requires version number checking for both control TLVs and feature TLVs.

DCBX interacts with other protocols and features as follows:

- *LLDP* — LLDP is run in parallel with other Layer 2 protocols such as RSTP and LACP. DCBX is built on the LLDP infrastructure to communicate capabilities supported between link partners. The DCBX protocol and feature TLVs are treated as a superset of the LLDP standard.
- *QoS management* — DCBX capabilities exchanged with a link partner are passed down to the QoS management entity to set up the Brocade VDX hardware to control the scheduling and priority-based flow control in the hardware.

The DCBX QoS standard is subdivided into two features sets, discussed below:

- Enhanced Transmission Selection on page 197
- Priority Flow Control on page 197

Enhanced Transmission Selection

In a converged network, different traffic types affect the network bandwidth differently. The purpose of Enhanced Transmission Selection (ETS) is to allocate bandwidth based on the different priority settings of the converged traffic. For example, Inter-process communications (IPC) traffic can use as much bandwidth as needed and there is no bandwidth check; LAN and SAN traffic share the remaining bandwidth. The table below displays three traffic groups: IPC, LAN, and SAN. ETS allocates the bandwidth based on traffic type and also assigns a priority to the three traffic types as follows: Priority 7 traffic is mapped to priority group 0 which does not get a bandwidth check, priority 2 and priority 3 are mapped to priority group 1, priorities 6, 5, 4, 1 and 0 are mapped to priority group 2.

The priority settings shown in the following table are translated to priority groups in the Brocade VDX hardware.

TABLE 24 ETS priority grouping of IPC, LAN, and SAN traffic

Priority	Priority group	Bandwidth check
7	0	No
6	2	Yes
5	2	Yes
4	2	Yes
3	1	Yes
2	1	Yes
1	2	Yes
0	2	Yes

Priority Flow Control

With Priority Flow Control (PFC), it is important to provide lossless frame delivery for certain traffic classes while maintaining existing LAN behavior for other traffic classes on the converged link. This differs from the traditional 802.3 PAUSE type of flow control where the pause affects all traffic on an interface.

PFC is defined by a one-byte bitmap. Each bit position stands for a user priority. If a bit is set, the flow control is enabled in both directions (Rx and Tx).

> **NOTE**
> When PFC is enabled, the Brocade VDX 6740 series platforms support up to three PGIDs with the execution of **cee-map default** . By default, PGID 1 (with TC3) and PGID 15.0 (for network control traffic) are enabled when PFC is enabled.

LLDP configuration guidelines and restrictions

Follow these LLDP configuration guidelines and restrictions when configuring LLDP:

- Brocade's implementation of LLDP supports Brocade-specific TLV exchange in addition to the standard LLDP information.
- Mandatory TLVs are always advertised.
- The exchange of LLDP link-level parameters is transparent to the other Layer 2 protocols. The LLDP link-level parameters are reported by LLDP to other interested protocols.

  NOTE
  DCBX configuration simply involves configuring DCBX-related TLVs to be advertised. Detailed information is provided in Configuring and managing LLDP on page 198.

Configuring and managing LLDP

The following sections discuss working with the Link Layer Discovery Protocol (LLDP) on Brocade devices.

Understanding the default LLDP

The following table lists the default LLDP configuration. Consider this when making changes to the defaults.

TABLE 25 Default LLDP configuration

Parameter	Default setting
LLDP global state	Enabled
LLDP receive	Enabled
LLDP transmit	Enabled
Transmission frequency of LLDP updates	30 seconds
Hold time for receiving devices before discarding	120 seconds
DCBX-related TLVs to be advertised	dcbx-tlv

Enabling LLDP globally

The **protocol lldp** command enables LLDP globally on all interfaces unless it has been specifically disabled on an interface, or the global LLDP disable command has been executed. LLDP is globally enabled by default.

To enable LLDP globally, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to change to global configuration mode.

   ```
   switch# configure terminal
   ```

2. Enter LLDP configuration mode.

   ```
   switch(config)# protocol lldp
   ```

3. Enter the **copy running-config startup-config** command to save your configuration changes.

Disabling LLDP globally

LLDP is enabled globally by default. These instructions allow you to disable LLDP globally without changing any other aspect of the LLDP configuration.

> NOTE
> The **disable** command executed in LLDP protocol configuration mode disables LLDP globally. To re-enable LLDP, refer to Enabling LLDP globally on page 198.

To disable LLDP globally, perform the following steps from global configuration mode.

1. Enter the **protocol lldp** command to enter protocol configuration mode.

```
switch(config)# protocol lldp
```

2. Enter the **disable** command to disable LLDP globally.

```
switch(conf-lldp)# disable
```

Resetting LLDP globally

The **no protocol lldp** command returns all configuration settings made using the protocol LLDP commands to their default settings.

To reset LLDP globally, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Reset LLDP globally.

```
switch(config)# no protocol lldp
```

Configuring LLDP global command options

After entering the **protocol lldp** command from global configuration mode, you are in LLDP configuration mode, which is designated with the switch(conf-lldp)# prompt. Using the keywords in this mode, you can set nondefault parameter values that apply globally to all interfaces.

Specifying a system name for the Brocade VDX hardware

The global system name for LLDP is useful for differentiating between switches. By default, the "host-name" from the chassis/entity management information base is used. By specifying a descriptive system name, you will find it easier to configure the switch for LLDP.

To specify a global system name for the Brocade VDX hardware, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Specify an LLDP system name for the DCB switch.

```
switch(conf-lldp)# system-name Brocade_Alpha
Brocade_Alpha(conf-lldp)#
```

Specifying an LLDP system description for the Brocade VDX hardware

NOTE
Brocade recommends you use the operating system version for the description or use the description from the chassis/entity management information base (MIB). Do not use special characters, such as #$!@, as part of the system name and description.

To specify an LLDP system description for the Brocade VDX hardware, perform the following steps from privileged EXEC mode. The system description is seen by neighboring switches.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Specify a system description for the Brocade VDX hardware.

```
switch(conf-lldp)# system-description IT_1.6.2_LLDP_01
```

Specifying a user description for LLDP

To specify a user description for LLDP, perform the following steps from privileged EXEC mode. This description is for network administrative purposes and is not seen by neighboring switches.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Specify a user description for LLDP.

```
switch(conf-lldp)# description Brocade-LLDP-installed-jan-25
```

Enabling and disabling the receiving and transmitting of LLDP frames

By default both transmit and receive for LLDP frames is enabled. To enable or disable the receiving (rx) and transmitting (tx) of LLDP frames, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Enter the **mode** command to do one of the following:
   - Enable only receiving of LLDP frames:

```
switch(conf-lldp)# mode rx
```

   - Enable only transmitting of LLDP frames:

```
switch(conf-lldp)# mode tx
```

   - Enable both transmit and receive modes.

```
switch(conf-lldp)# no mode
```

Configuring the transmit frequency of LLDP frames

To configure the transmit frequency of LLDP frames, perform the following steps from privileged EXEC mode. The default is 30 seconds.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Configure the transmit frequency of LLDP frames.

```
switch(conf-lldp)# hello 45
```

Configuring the hold time for receiving devices

To configure the hold time for receiving devices, perform the following steps from privileged EXEC mode. This configures the number of consecutive LLDP hello packets that can be missed before removing the neighbor information. The default is 4.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Configure the hold time for receiving devices.

```
switch(conf-lldp)# multiplier 6
```

Advertising the optional LLDP TLVs

To advertise the optional LLDP TLVs, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Advertise the optional LLDP TLVs.

```
switch(conf-lldp)# advertise optional-tlv management-address port-description system-capabilities
system-name system-description
```

Configuring the advertisement of LLDP DCBX-related TLVs

For a switch in Brocade VCS Fabric mode the following TLVs are advertised by default:

- dcbx-tlv
- dcbx-fcoe-app-tlv
- dcbx-fcoe-logical-link-tlv

To configure the LLDP DCBX-related TLVs to be advertised, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Advertise the LLDP DCBX-related TLVs by using one of these commands:
   - `switch(conf-lldp)# advertise dcbx-fcoe-app-tlv`
   - `switch(conf-lldp)# advertise dcbx-fcoe-logical-link-tlv`
   - `switch(conf-lldp)# advertise dcbx-tlv`

Configuring LLDP profiles

You can configure up to 64 profiles on a switch. Entering **no profile** *name* deletes the entire profile.

To configure LLDP profiles, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Configure the profile name.

```
switch(conf-lldp)# profile UK_LLDP_IT
```

4. Specify a description for the profile.

```
switch(conf-lldp-profile-UK_LLDP_IT)#description standard_profile_by_Jane
```

5. Enable the transmitting and receiving of LLDP frames.

```
switch(conf-lldp-profile-UK_LLDP_IT)# no mode
```

6. Configure the transmission frequency of LLDP updates.

```
switch(conf-lldp-profile-UK_LLDP_IT)# hello 10
```

7. Configure the hold time for receiving devices.

```
switch(conf-lldp-profile-UK_LLDP_IT)# multiplier 2
```

8. Advertise the optional LLDP TLVs.

```
switch(conf-lldp)# advertise optional-tlv management-address port-description
        system-capabilities system-name system-description
```

9. Advertise the LLDP DCBX-related TLVs.

```
switch(conf-lldp-profile-UK_LLDP_IT)# advertise advertise dcbx-tlv
switch(conf-lldp-profile-UK_LLDP_IT)# advertise dcbx-fcoe-logical-link-tlv
switch(conf-lldp-profile-UK_LLDP_IT)# advertise dcbx-fcoe-app-tlv
switch(conf-lldp-profile-UK_LLDP_IT)# advertise dcbx-iscsi-app-tlv
```

   NOTE

   Brocade recommends against advertising dot1.tlv and dot3.tlv LLDPs if your network contains CNAs from non-Brocade vendors, as doing so may cause functionality problems.

10. Return to privileged EXEC mode.

```
switch(conf-lldp-profile-UK_LLDP_IT)# end
```

11. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch(conf-lldp-profile-UK_LLDP_IT)# end
switch# copy running-config startup-config
```

Configuring iSCSI priority

The iSCSI TLV is used only to advertise the iSCSI traffic configuration parameters to the attached CEE enabled servers and targets. No verification or enforcement of the usage of the advertised parameters by the iSCSI server or target is done by the switch. The iSCSI priority setting is used to configure the priority to be advertised in the DCBx iSCSI TLV.

To configure the iSCSI priority, perform the following steps from privileged EXEC mode.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter LLDP configuration mode.

```
switch(config)# protocol lldp
```

3. Configure the iSCSI priority.

```
switch(conf-lldp)# iscsi-priority 4
```

NOTE

The default iscsi-priority is 4 and does not display unless you change the iscsi-priority to a different value.

4. Advertise the TLV.

```
switch (conf-lldp)# advertise dcbx-iscsi-app-tlv
```

Configuring the iSCSI profile

You can configure an iSCSI profile to be applied to individual interfaces. However, the priority bit must be set manually for each interface. Using the **no profile** *name* command deletes the entire profile.

To configure iSCSI profiles, perform the following steps from privileged EXEC mode.

1. Configure the CEE map, if it has not already been created. For information on the **cee-map** command structure, refer to the *Network OS Command Reference*.

```
switch(config)# cee-map default
switch(conf-ceemap)# priority-group-table 1 weight 50 pfc
switch(conf-ceemap)# priority-group-table 2 weight 30 pfc on
switch(conf-ceemap)# priority-group-table 3 weight 20 pfc on
switch(conf-ceemap)# priority-table 1 1 1 1 2 3 1 1
```

The **priority-table** command syntax is as follows:

```
priority-table PGID0 PGID1 PGID2 PGID3 PGID4 PGID5 PGID6 PGID7
```

For all PGID values, the PGID value range is 0 through 7 for the DWRR Priority Group, and 15.0 through 15.7 for the Strict Priority Group. The PGID value and the CoS value are equivalent, so that specifying PGID0 sets the Priority Group ID for all packets with CoS = 0, specifying PGID1 sets the Priority Group ID for all packets with CoS = 1, all the way through specifying PGID7, which sets the Priority Group ID for all packets with CoS = 7.

Priority-Table in CEE map configuration requires that PGID 15.0 is dedicated for CoS7. Because of this restriction, make sure that PGID15.0 is configured only as the last parameter for Priority-Table configuration.

An explanation of syntax "priority-table 1 2 2 2 2 2 2 15.0" is as follows:

This shows the definition of a CEE Map with Priority to Priority Group mapping of CoS=1, CoS=2, CoS=3, CoS=4, CoS=5, and CoS=6 to a DWRR Priority Group ID of 2, and CoS=0 to a Priority Group ID of 1, and CoS=7 to a Strict Priority Group.

This is one way to provision the CEE Priority to Priority Group Table, which maps each of the eight ingress CoS into a Priority Group.

In VCS mode, traffic classes are either all strict priorities (802.1Q default) or a combination of strict and DWRR traffic classes.

2. Enter LLDP configuration mode.

```
switch(conf-ceemap)# protocol lldp
```

3. Create an LLDP profile for iSCSI.

```
switch(conf-lldp)# profile iscsi_config
```

4. Advertise the iSCSI TLV.

```
switch(conf-lldp-profile-iscsi_config)# advertise dcbx-iscsi-app-tlv
```

5. Enter configuration mode for the specific interface.

```
switch (conf-lldp-profile-iscsi_config)# interface te 5/0/1
```

6. Apply the CEE provisioning map to the interface.

```
switch(conf-if-te-5/0/1)# cee default
```

7. Apply the LLDP profile you created for iSCSI.

```
switch(conf-if-te-5/0/1)# lldp profile iscsi_config
```

8. Set the iSCSI priority bits for the interface.

```
switch(conf-if-te-5/0/1)# lldp iscsi-priority 4
```

9. Repeat steps 5 through 8 for additional interfaces.

Configuring LLDP interface-level command options

Only one LLDP profile can be assigned to an interface. If you do not use the **lldp profile** option at the interface level, the global configuration is used on the interface. If there are no global configuration values defined, the global default values are used.

To configure LLDP interface-level command options, perform the following steps from privileged EXEC mode.

1. Enter the **interface** command, specifying the DCB interface type and RBridge/slot/port.

```
switch(config)# interface tengigabitethernet 5/0/10
```

2. Apply an LLDP profile to the interface.

```
switch(conf-if-te-5/0/10)# lldp profile network_standard
```

3. Return to privileged EXEC mode.

```
switch(conf-if-te-5/0/10)# end
```

4. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Displaying LLDP-related information

To display LLDP-related information, perform one or more of the following steps from privileged EXEC mode.

1. To display LLDP general information, use the **show lldp** command.

```
switch# show lldp
```

2. To display LLDP interface-related information, use the **show lldp** command .

```
switch# show lldp interface tengigabitethernet 22/0/1
LLDP information for gi 22/0/1
  State:                   Enabled
  Mode:                    Receive/Transmit
  Advertise Transmitted:   30 seconds
  Hold time for advertise: 120 seconds
  Re-init Delay Timer:     2 seconds
  Tx Delay Timer:          1 seconds tengigabitethernet
  DCBX Version :           CEE
  Auto-Sense :             Yes
  Transmit TLVs:           Chassis ID          Port ID
                           TTL                 IEEE DCBX
                           DCBX FCoE App       DCBX FCoE Logical Link
                           Link Prim           Brocade Link
  DCBX FCoE Priority Bits: 0x8
```

3. To display LLDP neighbor-related information, use the **show lldp neighbors** command.

```
switch# show lldp neighbors interface tengigabitethernet 22/0/1 detail
Neighbors for Interface Te 22/0/1

MANDATORY TLVs
===============
Local Interface: Te 22/0/1  (Local Interface MAC: 0027.f854.501e)
Remote Interface: TenGigabitEthernet 3/0/1 (Remote Interface MAC: 0005.334b.7198)
Dead Interval: 120 secs
Remaining Life : 117 secs
Chassis ID: 0005.334b.7173
LLDP PDU Transmitted: 1165  Received: 1164

OPTIONAL TLVs
==============
DCBX TLVs
==============
Version : CEE
DCBX Ctrl OperVersion: 0  MaxVersion: 0  SeqNo: 2 AckNo: 2
DCBX ETS OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
Enhanced Transmission Selection (ETS)
  Priority-Group ID Map:
    Priority : 0  1  2  3  4  5  6  7
    Group ID : 0  0  0  0  0  0  0  0
  Group ID Bandwidth Map:
    Group ID  :   0   1   2   3   4   5   6   7
    Percentage:   0   0   0   0   0   0   0   0
    Number of Traffic Classes supported: 8
DCBX PFC OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
Priority-based Flow Control (PFC)
     Enabled Priorities: none
     Number of Traffic Class PFC supported: 8
FCoE App OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
FCoE Application Protocol
     User Priorities: none
FCoE LLS OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
FCoE Logic Link Status: Down
LAN LLS OperVersion: 0  MaxVersion: 0  Enabled: 1  Willing: 0 Error: 0
LAN  Logic Link Status: Up
```

Clearing LLDP-related information

To clear LLDP-related information, perform the following steps from privileged EXEC mode.

1. Use the **clear** command to clear LLDP neighbor information.

```
switch# clear lldp neighbors interface tengigabitethernet 0/1
```

2. Use the **clear** command to clear LLDP statistics.

```
switch# clear lldp statistics interface tengigabitethernet 0/1
```

QoS

QoS overview

Quality of Service (QoS) provides you with the capability to control how the traffic is moved from switch to switch. In a network that has different types of traffic with different needs (specified by Class of Service, or CoS), the goal of QoS is to provide each traffic type with a virtual pipe. FCoE uses traffic class mapping, scheduling, and flow control to provide quality of service.

Traffic running through the switches can be classified as either multicast traffic or unicast traffic. Multicast traffic has a single source but multiple destinations. Unicast traffic has a single source with a single destination. With all this traffic going through inbound and outbound ports, QoS can be set based on egress port and priority level of the CoS.

QoS can also be set on interfaces where the end-station knows how to mark traffic with QoS and it lies with the same trusted interfaces. An untrusted interface occurs when the end-station is untrusted and is at the administrative boundaries.

QoS features

The principal QoS features are as follows:

- Rewriting. Rewriting or marking a frame allows for overriding header fields such as the priority and VLAN ID. Refer to Rewriting on page 217 for more information.
- Queueing. Queueing provides temporary storage for frames while waiting for transmission. Queues are selected based on ingress ports, egress ports, and configured user priority level. Refer to Queueing on page 207 for more information.
- Congestion control. When queues begin filling up and all buffering is exhausted, frames are dropped. This has a detrimental effect on application throughput. Congestion control techniques are used to reduce the risk of queue overruns without adversely affecting network throughput. Congestion control features include IEEE 802.3x Ethernet Pause, Tail Drop, Ethernet Priority Flow Control (PFC), and Random Early Detect (RED). Refer to Congestion control on page 208 for more information.
- Multicast rate limiting. Many multicast applications cannot be adapted for congestion control techniques and the replication of frames by switching devices can exacerbate this problem. Multicast rate limiting controls frame replication to minimize the impact of multicast traffic. This feature is called BUM Storm Control on Brocade VDX 8770-4, VDX 8770-8, and later platforms. Refer to Multicast rate limiting for more information.
- BUM storm control. A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. BUM storm control on page 228 allows you to limit the amount of broadcast, unknown unicast, and multicast (BUM) traffic admitted to the system to prevent disruptions on Layer 2 physical ports. All traffic received at a physical port in excess of a configured maximum rate for BUM traffic will be discarded. You can also specify whether to shut down an interface if the maximum rate has been exceeded within a 5-second sampling period and receive a LOG indication for the disabled interface. This feature is supported on Brocade VDX 8770, VDX 6740, VDX 6740-T, VDX 6940 series, and VDX 2746 platforms.
- Data Center Bridging. DCB describes an enhanced Ethernet that will enable convergence of various applications in data centers (LAN, SAN, and IPC) onto a single interconnect technology.

Queueing

Queue selection begins by mapping an incoming frame to a configured user priority, then each user-priority mapping is assigned to one of the switch's eight unicast traffic class queues or one of the eight multicast traffic class queues.

User-priority mapping

There are several ways an incoming frame can be mapped into a user-priority. If the neighboring devices are untrusted or unable to properly set QoS, then the interface is considered untrusted. All traffic must be user-priority mapped using explicit policies for the interface to be trusted; if it is not mapped in this way, the IEEE 802.1Q default-priority mapping is used. If an interface is trusted to have QoS set then the CoS header field can be interpreted.

NOTE
The user priority mapping discussed in this chapter applies to both unicast and multicast traffic.

Congestion control

Queues can begin filling up for a number of reasons, such as over-subscription of a link or backpressure from a downstream device. Sustained, large queue buildups generally indicate congestion in the network and can affect application performance through increased queuing delays and frame loss.

Congestion control covers features that define how the system responds when congestion occurs or active measures taken to prevent the network from entering a congested state.

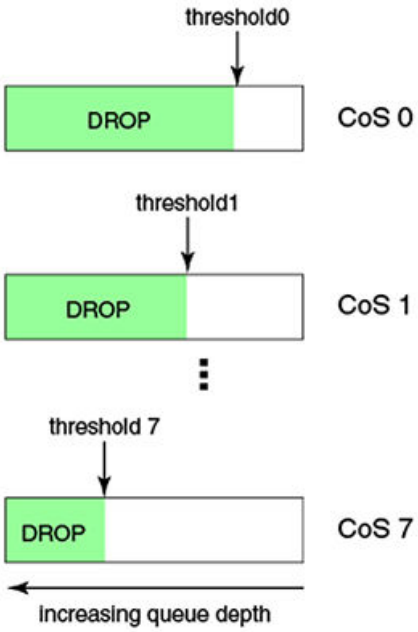
NOTE
You cannot configure CoS thresholds and multicast tail drop on Brocade VDX 8770-4 and VDX 8770-8 platforms. Weighted Random Early Detection (WRED) is supported only on Brocade VDX 6740, VDX 6940, VDX 8770-4, and VDX 8770-8 platforms.

Tail drop

Tail drop queuing is the most basic form of congestion control. Frames are queued in FIFO order and queue buildup can continue until all buffer memory is exhausted. This is the default behavior when no additional QoS has been configured.

The basic tail drop algorithm does not have any knowledge of multiple priorities and per traffic class drop thresholds can be associated with a queue to address this. When the queue depth breaches a threshold, then any frame arriving with the associated priority value will be dropped. The figure below illustrates how you can utilize this feature to ensure that lower-priority traffic cannot totally consume the full buffer memory.

FIGURE 34 Queue depth



Thresholds can also be used to bound the maximum queuing delay for each traffic class. Additionally, if the sum of the thresholds for a port is set below 100 percent of the buffer memory, then you can also ensure that a single port does not monopolize the entire shared memory pool allocated to the port. The tail drop algorithm can be extended to support per-priority drop thresholds. When the ingress port CoS queue depth breaches a threshold, then any frame arriving with the associated priority value will be dropped.

Weighted Random Early Detection

NOTE
This feature is only supported on the Brocade VDX 8770 series, VDX 6740 series, VDX 6940 series, and VDX 2746.

Traditionally, Weighted Random Early Detection (WRED) is used for TCP traffic streams, which are generally more aggressive, as well as reactive, to network drops. If WRED is not configured, queues build up at the switch and become full, resulting in tail drop. Tail drop situations can cause head-of-line blocking issues at the switch, which is not desirable. By configuring WRED, you set a probability for dropping packets before traffic in the queue reaches a specific threshold. This allows congestion to ease more gradually, avoids retransmit synchronization, resolves "bursty" TCP connections during congestion conditions, and controls packet latency.

Configure WRED using the following parameters:

- WRED profile identification (0-384)
- Minimum threshold of a queue (0-100%)
- Maximum threshold of a queue (0-100%)
- Drop probability (0-100%)

NOTE
Beginning with Network OS 5.0.0, the maximum number of WRED profiles at the system level is 3 on the Brocade VDX 8770 series, 16 on the Brocade VDX 6740 series and the VDX 2746, and 15 on the VDX 6940 series.

The ASIC driver maps the configured minimum and maximum percentages to the actual queue size in bytes, depending on the bandwidth of the port (buffers are allocated to a port according to port speed). When buffers in the queue build up to the set minimum threshold, packets being queued are randomly dropped. The drop probability parameter defines the randomness of the drops. When the queues exceed the minimum threshold, packets are dropped according to the configured drop probability value. When the queue buffers exceed the set maximum threshold, packets are dropped with 100% probability. The higher the probability set, the more likely packets will be dropped when the minimum percentage is reached.

You can also map a specific CoS priority value (0 through 7) to a specific WRED profile.

Configuring dynamic buffer sharing

If there is bursty, lossy traffic for certain flows in the system, you can borrow the buffers from less bursty flows, in order to reduce the traffic loss. The **qos** command is used to configure the egress or ingress queue limit, such as the maximum number of kilobytes of data that can be queued in the egress or ingress queue. The **tx-queue** keyword controls the egress, and the **rcv-queue** keyword controls the ingress.

This command only functions on the Brocade VDX 6740 series, VDX 6940 series, and VDX 2746. This configuration is applied on individual RBridges.

To configure dynamic buffer sharing, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Enter RBridge ID configuration mode.

```
switch(config)# rbridge-id 154
```

3. Configure the dynamic buffer sharing. The following example sets the egress limit to 256 kilobytes. (The default is 512 kilobytes.)

```
(config-rbridge-id-154)# qos tx-queue limit 256
```

   The following example sets the ingress limit to 1024 kilobytes. (The default is 285 kilobytes.)

```
(config-rbridge-id-154)# qos rcv-queue limit 1024
```

4. Return to privileged EXEC mode.

```
(config-rbridge-id-154)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Enabling drop logging

Use this procedure to enable RASlog messages for dropped data on an interface.

NOTE

The following drop types are logged:

- Brocade VDX 2746, VDX 6740 series, and VDX 6940 series: Random Early Detect (RED) and tail drops
- Brocade VDX 8770 (internal port interface ASICs): tail drops only

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. Enter the **interface** command to access interface subtype configuration mode for the interface on which you are implementing drop logging.

```
device(config)# interface tengigabitethernet 2/2/1
```

3. Enter the **qos drop-monitor enable** command.

```
device(conf-if-te-2/2/1)# qos drop-monitor enable
```

Ethernet Pause

Ethernet Pause is an IEEE 802.3 standard flow-control mechanism for back pressuring a neighboring device. Pause messages are sent by utilizing the optional MAC control sublayer. A PAUSE frame contains a 2-byte pause number, which states the length of the pause in units of 512 bits. When a device receives a PAUSE frame, it must stop sending any data on the interface for the specified length of time, once it completes the transmission of any frame in progress. You can use this feature to reduce Ethernet frame losses by using a standardized mechanism. However, the pause mechanism does not have the ability to selectively back-pressure data sources multiple hops away, or to exert any control per VLAN or per priority, so it is disruptive to all traffic on the link.

Ethernet Pause features

Ethernet Pause includes the following features:

- All configuration parameters can be specified independently per interface.
- Pause On/Off can be specified independently for TX and RX directions. No support is provided for disabling autonegotiation.
- Pause generation is based on input (receive) queueing. Queue levels are tracked per input port. When the instantaneous queue depth crosses the high-water mark, then a PAUSE frame is generated. If any additional frames are received and the queue length is still above the low-water mark, then additional PAUSE frames are generated. Once the queue length drops below the low-water mark, then the generation of PAUSE frames ceases.
- A PAUSE frame that is received and processed halts transmission of the output queues associated with the port for the duration specified in the PAUSE frame.

1-Gbps pause negotiation

When a 1-Gbps local port is already online, and the **qos flowcontrol** command is issued, the pause settings take effect immediately on that local port. However, when the link is toggled, pause is renegotiated. The local port will advertise the most recent **qos flowcontrol** settings. After autonegotiation completes, the local port pause settings may change, depending on the outcome of the pause negotiation, per 802.3 Clause 28B, as shown in the table below.

TABLE 26 Pause negotiation results

Advertised LOCAL cfg	Advertised REMOTE cfg	Negotiated result
Rx=off Tx=on	Rx=on Tx=on	asymmetrical: LOCAL Tx=on –> pause –> REMOTE Rx=on
Rx=on Tx=on	Rx=off Tx=on	asymmetrical: LOCAL Rx=on <– pause <– REMOTE Tx=on
Rx=on Tx=n/a	Rx=on Tx=n/a	symmetrical: LOCAL Tx/Rx=on <– pause –> REMOTE Tx/Rx=on
Rx=n/a Tx=n/a	Rx=off Tx=off	disable pause both sides

Ethernet Priority Flow Control

Ethernet Priority Flow Control (PFC) is a basic extension of Ethernet Pause. The Pause MAC control message is extended with eight 2-byte pause numbers and a bitmask to indicate which values are valid. Each pause number is interpreted identically to the base Pause protocol; however, each number is applied to the corresponding Ethernet priority/class level. For example, the Pause number 0 applies to priority zero, Pause number 1 applies to priority one, and so on. This addresses one shortcoming of the Ethernet Pause mechanism, which is disruptive to all traffic on the link. However, it still suffers from the other Ethernet Pause limitations.

> NOTE
> The Brocade VDX 6740 series and VDX 6940 series switches support a maximum of two PFC (lossles) CoS profiles on an interface. If any of these switches is present in a VCS cluster, the user should not create more than two lossless CoS profiles in the CEE configuration.

Ethernet Priority Flow Control includes the following features:

- Everything operates exactly as in Ethernet Pause described above, except there are eight high-water and low-water thresholds for each input port. This means queue levels are tracked per input port plus priority.
- Pause On/Off can be specified independently for TX and RX directions per priority.
- Pause time programmed into Ethernet MAC is a single value covering all priorities.
- Both ends of a link must be configured identically for Ethernet Pause or Ethernet Priority Flow Control because they are incompatible.

Scheduling

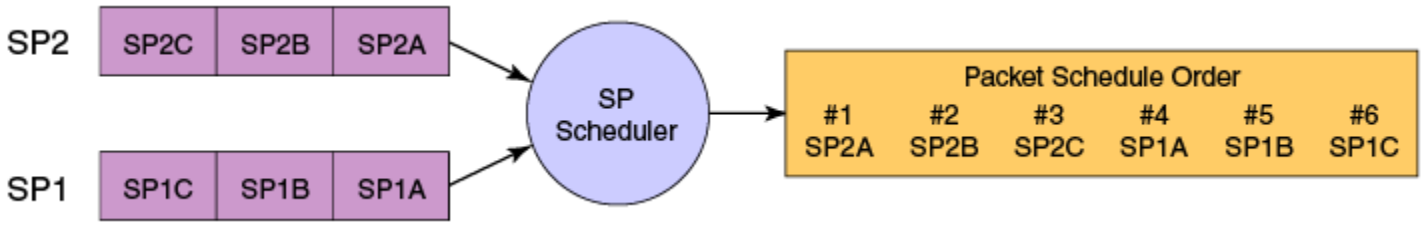
Scheduling arbitrates among multiple queues waiting to transmit a frame. The Brocade switch supports both Strict Priority (SP) and Deficit Weighted Round Robin (DWRR) scheduling algorithms. Also supported is the flexible selection of the number of traffic classes using SP-to-DWRR. When there are multiple queues for the same traffic class, then scheduling takes these equal-priority queues into consideration.

Strict priority scheduling

Strict priority scheduling is used to facilitate support for latency-sensitive traffic. A strict priority scheduler drains all frames queued in the highest-priority queue before continuing on to service lower-priority traffic classes. A danger with this type of service is that a queue can potentially starve out lower-priority traffic classes.

The following figure displays the frame scheduling order for an SP scheduler servicing two SP queues. The higher-numbered queue, SP2, has a higher priority.

FIGURE 35 Strict priority schedule — two queues

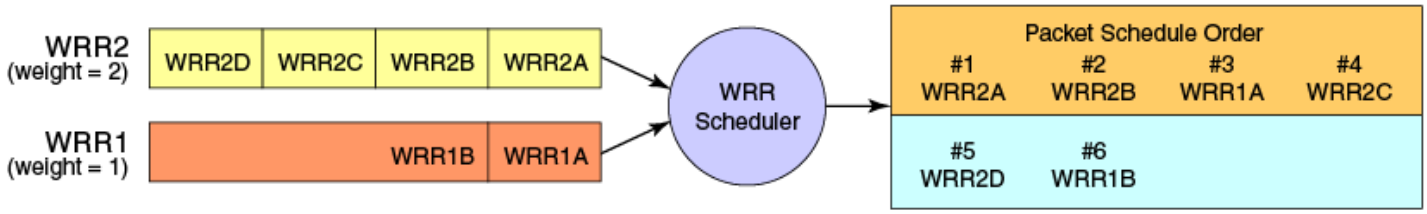


Weighted Round Robin scheduling

Weighted Round Robin (WRR) scheduling is used to facilitate controlled sharing of the network bandwidth. WRR assigns a weight to each queue; that value is then used to determine the amount of bandwidth allocated to the queue. The round robin aspect of the scheduling allows each queue to be serviced in a set order, sending a limited amount of data before moving onto the next queue and cycling back to the highest-priority queue after the lowest-priority queue is serviced.

The following figure displays the frame scheduling order for a WRR scheduler servicing two WRR queues. The higher-numbered queue is considered higher priority (WRR2), and the weights indicate the network bandwidth should be allocated in a 2:1 ratio between the two queues. In this figure WRR2 receives 66 percent of the bandwidth and WRR1 receives 33 percent. The WRR scheduler tracks the extra bandwidth used and subtracts it from the bandwidth allocation for the next cycle through the queues. In this way, the bandwidth utilization statistically matches the queue weights over longer time periods.

FIGURE 36 WRR schedule — two queues



Deficit Weighted Round Robin (DWRR) is an improved version of WRR. DWRR remembers the excess used when a queue goes over its bandwidth allocation and reduces the queue's bandwidth allocation in the subsequent rounds. This way the actual bandwidth usage is closer to the defined level when compared to WRR.

Traffic class scheduling policy

Traffic classes are numbered from 0 to 7, with higher-numbered traffic classes treated as having a higher priority. Brocade switches provide full flexibility in controlling the number of SP-to-WRR queues. The number of SP queues is specified as SP1 through 8, then the highest-priority traffic classes are configured for SP service and the remaining eight are WRR serviced. The supported scheduling configurations listed in the table below describes the set of scheduling configurations supported.

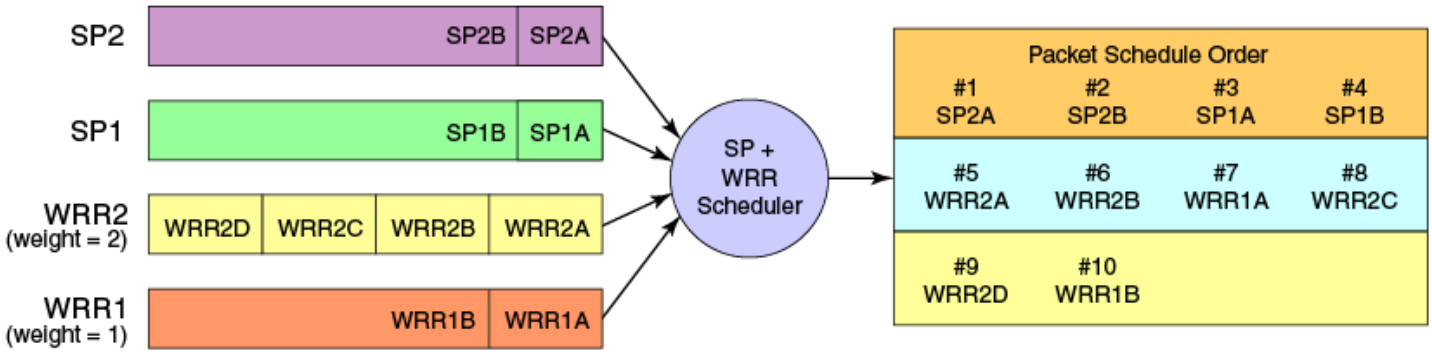
When you configure the QoS queue to use strict priority 4 (SP4), then traffic class 7 will use SP4, traffic class 6 will use SP3, and so on down the list. You use the strict priority mappings to control how the different traffic classes will be routed in the queue.

TABLE 27 Supported scheduling configurations

Traffic Class	SP0	SP1	SP2	SP3	SP4	SP5	SP6	SP8
7	WRR8	SP1	SP2	SP3	SP4	SP5	SP6	SP8
6	WRR7	WRR7	SP1	SP2	SP3	SP4	SP5	SP7
5	WRR6	WRR6	WRR6	SP1	SP2	SP3	SP4	SP6
4	WRR5	WRR5	WRR5	WRR5	SP1	SP2	SP3	SP5
3	WRR4	WRR4	WRR4	WRR4	WRR4	SP1	SP2	SP4
2	WRR3	WRR3	WRR3	WRR3	WRR3	WRR3	SP1	SP3
1	WRR2	WRR2	WRR2	WRR2	WRR2	WRR2	WRR2	SP2
0	WRR1	WRR1	WRR1	WRR1	WRR1	WRR1	WRR1	SP1

The figure below shows that extending the frame scheduler to a hybrid SP+WRR system is fairly straightforward. All SP queues are considered strictly higher priority than WRR so they are serviced first. Once all SP queues are drained, then the normal WRR scheduling behavior is applied to the non-empty WRR queues.

FIGURE 37 Strict priority and Weighted Round Robin scheduler



Multicast queue scheduling

The multicast traffic classes are numbered from 0 to 7; higher numbered traffic classes are considered higher priority. A fixed mapping from multicast traffic class to equivalent unicast traffic class is applied to select the queue scheduling behavior. The Multicast traffic class equivalence mapping table below presents the multicast traffic class with the equivalence mapping applied.

Once the multicast traffic class equivalence mapping has been applied, then scheduling and any scheduler configuration are inherited from the equivalent unicast traffic class. Refer to the table below for details on exact mapping equivalencies.

TABLE 28 Multicast traffic class equivalence mapping

Multicast traffic class	Equivalent unicast traffic class
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Unicast ingress and egress queueing utilizes a hybrid scheduler that simultaneously supports SP+WRR service and multiple physical queues with the same service level. Multicast adds additional multicast expansion queues. Because multicast traffic classes are equivalent to unicast service levels, they are treated exactly as their equivalent unicast service policies.

Data Center Bridging QoS

Data Center Bridging (DCB) QoS covers frame classification, priority and traffic class (queue) mapping, congestion control, and scheduling. Under the DCB provisioning model, all of these features are configured on the basis of two configuration tables, Priority Group Table and Priority Table.

The DCB Priority Group Table defines each Priority Group ID (PGID) and its scheduling policy (Strict Priority versus DWRR, DWRR weight, relative priority), and partially defines the congestion control (PFC) configuration. There are 16 rows in the DCB Priority Group Table.

The table below presents the default DCB Priority Group Table configuration.

TABLE 29 Default DCB Priority Group Table configuration

PGID	Bandwidth%	PFC
15.0	-	Y
15.1	-	N
15.2	-	N
15.3	-	N
15.4	-	N
15.5	-	N
15.6	-	N
15.7	-	N
0	0	N
1	0	Y
2	0	N
3	0	N
4	0	N
5	0	N
6	0	N
7	0	N

NOTE

Only a single CoS can be mapped to a PFC-enabled priority queue. The switch automatically maps the CoS number to the same TC number when PFC is enabled. The PGID can be anything from 0 through 7. If your configuration violates this restriction an error message displays and the Priority Group Table is set back to the default values. When the DCB map is applied, and the interface is connected to the CNA, only one Strict Priority PGID (PGID 15.0 through PGID 15.7) is allowed.

Strict Priority versus DWRR is derived directly from the PGID value. All PGIDs with prefix 15 receive Strict Priority scheduling policy, and all PGIDs in the range 0 through 7 receive DWRR scheduling policy. Relative priority between Priority Group is exactly the ordering of entries listed in the table, with PGID 15.0 being highest priority and PGID 7 being lowest priority. Congestion control configuration is partially specified by toggling the PFC column On or Off. This provides only partial configuration of congestion control because the set of priorities mapped to the Priority Group is not known, which leads into the DCB Priority Table.

The DCB Priority Table defines each CoS mapping to Priority Group, and completes PFC configuration. The table below shows an example of mapping in the DCB Priority Table.

TABLE 30 Example of mapping DCB priority table values

CoS	PGID
0	15.6
1	15.7
2	15.5
3	15.4
4	15.3
5	15.2

TABLE 30 Example of mapping DCB priority table values (continued)

CoS	PGID
6	15.1
7	15.0

Brocade VCS Fabric QoS

Brocade VCS Fabric QoS requires very little user configuration. The only options to modify are the fabric priority and the lossless priority.

Brocade VCS Fabric reserves a mapping priority and fabric priority of seven (7). Any traffic that enters the Brocade VCS Fabric cluster from upstream that is using the reserved priority value is automatically remapped to a lower priority.

Changing the mapping or fabric priority is not required. By default the values are set to zero (0) for both of the remapped priorities.

In Brocade VCS Fabric mode:

- All incoming priority 7 tagged packets are redefined to the default or user-defined value.
- Untagged control frames are counted in queue 7 (TC7).

All switches in the Brocade VCS Fabric cluster must have matching remapping priority values and the same priority-group-table values.

Restrictions for Layer 3 features in VCS mode

When the switch is in VCS mode, the lossless priority for carrying FCoE traffic and the fabric priority for carrying fabric traffic must be isolated from any Layer 3 QoS markings and classification. Therefore, specific restrictions apply to some Layer 3 DSCP QoS features when the switch is working in VCS mode:

The following are restrictions for using applicable Layer 3 DSCP-Traffic-Class map, DSCP-CoS map, and DSCP Trust features in VCS mode. Note that DSCP mutation maps and the WRED feature are not affected in VCS mode.

- DSCP trust is disabled in VCS mode as it is for CoS trust.
- There are no default DSCP maps in VCS mode.
- A nondefault DSCP-Traffic-Class map has the following restrictions:
  - A DSCP value cannot be classified to Traffic Class 7.
  - A DSCP value cannot be classified to a queue that carries lossless traffic (by default Traffic Class 3).
- A nondefault DSCP-CoS map has the following restrictions:
  - A DSCP value cannot be marked to CoS 7.
  - A DSCP value cannot be marked to lossless priority (by default CoS 3).
- Lossless priorities are identified through the CEE map.
- To enable DSCP based marking or classification, a nondefault DSCP-Traffic-Class map and a DSCP-CoS map have to be applied on the interface.
- To apply a DSCP-Traffic-Class or DSCP-CoS map to an interface, the CoS and Traffic Class values have to be remarked for lossless priorities. For example, when DSCP-Traffic-Class map "abcd" is created, it will have the default contents. When this map is applied to an interface, an error will display that the fabric and lossless priorities are used in the map and it cannot be applied on the interface.
- When a valid DSCP-Traffic-Class map and DSCP-CoS map are applied on the interface, then DSCP trust is enabled with the configured maps.

Port-based Policer

The port-based Policer feature controls the amount of bandwidth consumed by an individual flow or aggregate of flows by limiting the inbound and outbound traffic rate on an individual port according to criteria defined by the user. The Policer provides rate control by prioritizing or dropping ingress and egress packets classified according to a two-rate, three-color marking scheme defined by RFC 4115.

Rewriting

Rewriting a frame header field is typically performed by an edge device. Rewriting occurs on frames as they enter or exit a network because the neighboring device is untrusted, unable to mark the frame, or is using a different QoS mapping.

The frame rewriting rules set the Ethernet CoS and VLAN ID fields. Egress Ethernet CoS rewriting is based on the user-priority mapping derived for each frame as described later in the queueing section.

Port-based Policer features

The Policer supports the following features.

- A color-based priority mapping scheme for limiting traffic rates.
  - One-rate, two-color policing with "conform" color options. "Violate" color traffic will be dropped.
  - Two-rate, three-color policing with "conform" and "exceed" color options. "Violate" color traffic will be dropped.
- A policing option that allows packet headers to be modified for IP precedence.
- Policing options that allow packet headers to be modified for Class of Service (CoS).
- Policing options that allow packet headers to be modified for Differentiated Services Code Point (DSCP).
- Policing options that allow packets to be assigned to a traffic class (0-7).

Color-based priority

The following is the color-based priority mapping scheme for limiting traffic rate:

- Traffic flagged to the green or "conform" color priority conforms to the committed information rate (CIR) as defined by the *cir-rate* variable for the policy-map (refer to Policing parameters on page 217). This rate can be anything from 40000 bps to 100 Gbps.
- Traffic flagged as yellow or "exceed" exceeds the CIR, but conforms to the Excess Information Rate (EIR) defined by the *eir-rate* variable for the policy-map (refer to Policing parameters on page 217). This rate can be set from 0 through 100 Gbps.
- Traffic flagged as red or "violate" are not compared to CIR or EIR and will be dropped.

Using policing parameters, you can define metering rates, such as CIR and EIR, and actions for traffic flagged as conforming or exceeding the rates. As a simple example, traffic within the "conform" rate may be sent at a certain CoS priority, traffic flagged at the "exceed" rate may be sent at a lower priority, and traffic that violates the set rates can be dropped (default and only option).

Policing parameters

Policing parameters provide values for CIR, CBS, EIR, and EBS, for classifying traffic by a specific class for color-based priority mapping. They also specify specific actions to perform on traffic with a color-class priority, such as having packet DSCP priority, traffic class (internal queue assignment), or traffic class (internal queue assignment) set to specific values.

CIR and CBS

The Committed Information Rate (CIR) is the maximum number of bits that a port can receive or send during one-second over an interface. For CIR, there are two parameters that define the available traffic: CIR and the Committed Burst Size (CBS). The CIR represents a portion of the interface's total bandwidth expressed in bits per second (bps). It cannot be larger than the interface's total

bandwidth. CBS controls the bursty nature of the traffic. Traffic that does not use the configured CIR accumulates credits until the credits reach the configured CBS. These credits can be used when the rate temporarily exceeds the configured CIR. When credits are not available, the traffic is either dropped or subject to the policy set for the Excess Information Rate (EIR). The traffic limited by the CIR can have its priority, traffic class, and DSCP values changed.

CIR is mandatory policing parameter for configuring a class map.

```
cir cir-rate
```

The **cir** command defines the value of the CIR as the rate provided in the *cir-rate* variable. Acceptable values are in multiples of 40000 in the range 40000-100000000000 bps.

```
cbs cbs-size
```

The **cbs** command defines the value of the CBS as the rate provided in the *cbs-size* variable. Acceptable values are 1250-12500000000 bytes in increments of 1 byte.

EIR and EBS

The Excess Information Rate (EIR) provides an option for traffic that has exceeded the CIR. For EIR, there are two parameters that define the available traffic: the EIR and the Excess Burst Size (EBS). The EIR and EBS operate exactly like the CIR and CBS, except that they act only upon traffic that has been passed to the EIR because it could not be accommodated by the CIR. Like the CIR, the EIR provides an initial bandwidth allocation to accommodate inbound and outbound traffic. Like the CBS, the bandwidth available for burst traffic from the EBS is subject to the amount of bandwidth that is accumulated during periods when traffic allocated by the EIR policy is not used. When inbound or outbound traffic exceeds the bandwidth available (accumulated credits or tokens), it is be dropped. The traffic rate limited by the EIR can have its priority, traffic class, and DSCP values changed.

EIR and EBS parameters are optional policing parameters. If not set, they are considered disabled.

```
eir eir-rate
```

The **eir** parameter defines the value of the EIR as the rate provided in the *eir-rate* variable. Acceptable values are in multiples of 40000 in the range 0-100000000000 bps.

```
 ebs ebs-size
```

The **ebs** parameter defines the value of the EBS as the rate provided in the *ebs-size* variable. Acceptable values are 1250-12500000000 bytes in increments of 1 byte.

Parameters that apply actions to conform and exceed traffic

Following are policing parameters that apply actions to conform or exceed color traffic:

- **conform-set-dscp** *dscp-num*.

  The **conform-set-dscp** parameter specifies that traffic with bandwidth requirements within the rate configured for CIR will have its packet DSCP priority set to the value specified in the *dscp-num* variable. Acceptable values for *dscp-num* are 0–63.

- **conform-set-prec** *prec-num*.

  The **conform-set-prec** parameter specifies that traffic with bandwidth requirements within the rate configured for CIR will have its packet IP precedence value (first 3 bits of DSCP) set to the value in the *prec-num* variable. Acceptable values for *prec-num* are 0–7.

- **conform-set-tc** *trafficlass*.

  The **conform-set-tc** parameter specifies that traffic with bandwidth requirements within the rate configured for CIR will have its traffic class (internal queue assignment) set to the value in the *trafficlass* variable. Acceptable values for *trafficclass* are 0–7.

- **exceed-set-dscp** *dscp-num*.

The **exceed-set-dscp** parameter specifies that traffic with bandwidth requirements that exceeds the rate configured for CIR and sent to the EIR bucket will have its packet DSCP priority set to the value in the *dscp-num* variable. Acceptable values for *dscp-num* are 0–63.

- **exceed-set-prec** *prec-num*.

  The **exceed-set-prec** parameter specifies that traffic with bandwidth requirements that exceed the rate configured for CIR and sent to the EIR bucket will have its packet IP precedence set to the value in the *prec-num* variable. Acceptable values for *prec-num* are 0–7.

- **exceed-set-tc** *trafficclass*.

  The **exceed-set-tc** parameter specifies that traffic with bandwidth requirements that exceed the rate configured for CIR and is in the limit of what is configured for EIR will have its traffic class (internal queue assignment) set to the value in the *trafficlass* variable. Acceptable values for *trafficlass* are 0–7.

- **set-priority** *priority-mapname*.

  The **set-priority parameter** specifies the mapping used for setting QoS priority (802.1p priority) in the packet. The *priority-mapname* name variable should be same as configured for the priority-map (police-priority-map), which will have a set priority and color type (conform or exceed).

Policer considerations and limitations

Consider the topics discussed below when configuring the port-based Policer feature.

Best practices for Policer

Follow these best practices when configuring the port-based Policer feature:

- Avoid mapping lossy priority to lossless priority in conform and exceed CoS maps.
- Configure rate (CIR or EIR) and burst size (CBS or EBS) based on interface speed.
- Set conform and exceed token count (Tc) to the same values to avoid any reordering issues.

Configuration rules and considerations for Policer

The following are rules for configuring maps and using policing parameters for the Policer feature:

- A policy-map, class map, priority-map name must be unique among all maps of that type.
- A policy-map is not supported on an ISL port.
- A Policer name must begin with a-z, or A-Z. You can use underscore, hyphen, and numeric values 0-9 except as the first character.
- You cannot delete a policy-map, class map, or priority-map if is active on the interface.
- You cannot delete a class map from a policy-map when the policy-map is active on the interface.
- Configure **CIR** and **EIR** in multiples of 40000 bps.
- Percentage as a rate limit is not supported,
- Policer actions are applicable only to data traffic. Control traffic, FCoE, and internal VLAN traffic is not subjected to policing.
- The egress Policer can overwrite ingress Policer results such as CoS mapping and DSCP mapping.
- If a policy-map is applied to an interface and no Policer attributes are present in that policy-map, then ingress and egress packets on that interface is marked as green (conforming).
- If the configured CBS value is less than 2*MTU value, then 2*MTU is programmed as the CBS in the hardware. For example, if you configure CBS at 4000 bytes and the MTU on an interface is 3000 bytes, when a policy-map is applied on this interface, the CBS programmed in the hardware is 2*MTU (6000 bytes).

- If CBS and EBS values are not configured, then these values are derived from CIR and EIR values, respectively. Burst size calculation is as follows: `Burst size (cbs or ebs) = 1.2*information rate (CIR/EIR)/8`
- If you do not configure EIR and EBS, then the single-rate, two-color scheme is applied (packets are marked as either green or red).
- You must configure rate limit threshold values on an interface based on interface speed. No validation is performed for user-configured values against interface speed.

Limitations for Policer

- The incremental step size for CIR or EIR is set to 40000 bps.
- The Policer operates in color-blind mode. In other words, color is evaluated at ingress and egress Policers independently. This may result in packets that are marked as yellow in the inbound Policer to be evaluated as green at the outbound Policer, depending on Policer settings.
- Because inbound queue scheduling is performed before outbound policing, setting traffic class (**set-conform-tc** or **set-exceed-tc**) based on policing results does not affect packet forwarding at the outbound side.
- Packets drops caused by any action other than ACLs are included in Policer counters.
- Layer 3 control packets are policed at the outbound side.
- Policing is enabled on lossless priorities at the outbound side.
- Policing is not enabled for control traffic that is trapped or dropped.

Considerations for vLAGs with Policer

Because a virtual link aggregation group (vLAG) spans multiple switches, it is not possible to associate flows on each LAG member port to a common Policer. Instead, apply the same policy-map on individual member ports so that traffic flow on member ports is controlled by a Policer configured on that member port. The total rate-limit threshold value on a vLAG consists of the cumulative values of rate-limit thresholds on all member ports.

Lossless traffic with Policer

The following are considerations for lossless traffic:

- Policing is applicable only for lossy traffic. Lossless traffic should not get policed. For port-based policing, apply a policy-map to an interface even if PFC is configured on that interface. The CoS value (priority) on which PFC is applied is excluded from being policed.
- Remapped priority values should not include lossless priorities. Do not remap lossy traffic priorities to lossless traffic priorities and vice-versa.
- Policer attributes **conform-set-tc** and **exceed-set-tc** should not be set to a lossless traffic class.

Configuring QoS

The following sections discuss configuring QoS, including fundamentals, traffic class mapping, congestion control, rate limiting, BUM storm control, scheduling, DCB QoS, Brocade VCS Fabric QoS, policer functions, and Auto QoS.

Configuring QoS fundamentals

NOTE

Refer to User-priority mapping on page 208.

Understanding default user-priority mappings for untrusted interfaces

When Layer 2 QoS trust is set to **untrusted** , then the default is to map all Layer 2 switched traffic to the port default user priority value of 0 (best effort), unless the user priority is configured to a different value.

The following table presents the Layer 2 QoS **untrusted** user-priority generation table.

TABLE 31 Default priority value of untrusted interfaces

Incoming CoS	User Priority
0	port <user priority> (default 0)
1	port <user priority> (default 0)
2	port <user priority> (default 0)
3	port <user priority> (default 0)
4	port <user priority> (default 0)
5	port <user priority> (default 0)
6	port <user priority> (default 0)
7	port <user priority> (default 0)

NOTE

Nontagged Ethernet frames are interpreted as having an incoming CoS value of 0 (zero).

You can override the default user-priority mapping by applying explicit user-priority mappings.

When neighboring devices are trusted and able to properly set QoS, then Layer 2 QoS trust can be set to **CoS** and the IEEE 802.1Q default-priority mapping is applied.

The following table presents the Layer 2 CoS user-priority generation table conforming to 802.1Q default mapping. You can override this default user-priority table per port if you want to change (mutate) the CoS value.

TABLE 32 IEEE 802.1Q default priority mapping

Incoming CoS	User Priority
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Configuring QoS mappings

Consider the topics discussed below when configuring the QoS mappings.

Configuring user-priority mappings

To configure user-priority mappings, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface.

```
switch(config)# interface tengigabitethernet 1/2/2
```

3. Configure the interface to priority 3.

```
switch(conf-if-te-1/2/2)# qos cos 3
```

4. Return to privileged EXEC mode.

```
switch(conf-if-te-1/2/2)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Creating a CoS-to-CoS mutation QoS map

To create a CoS-to-CoS mutation, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Create the CoS-to-CoS mutation QoS map. In this example "test" is the map name.

```
switch(config)# qos map cos-mutation test 0 1 2 3 4 5 6 7
```

3. Enter the **do copy** command to save the *running-config* file to the *startup-config* file.

```
switch(config)# do copy running-config startup-config
```

Applying a CoS-to-CoS mutation QoS map to an interface

To apply a CoS-to-CoS mutation QoS map, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface.

```
switch(config)# interface tengigabitethernet 2/1/2
```

3. Activate or apply changes made to the CoS-to-CoS mutation QoS map. In this example, "test" is the map name.

```
switch(conf-if-te-2/1/2)# qos cos-mutation test
```

NOTE

To deactivate the mutation map from an interface, enter **no qos cos-mutation** *name*.

4. Return to privileged EXEC mode.

```
switch(conf-if-te-2/1/2)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Verifying CoS-to-CoS mutation QoS mapping

To verify applied QoS maps, you can use one or both of the following options from global configuration mode.

- Verify the CoS mutation mapping for a specific map by using the **do show qos maps qos-mutation** command and the map name.

```
switch(config)# do show qos maps cos-mutation test
```

- Verify all QoS mapping by using the **do show qos maps** command with just the **cos-mutation** parameter only.

```
switch(config)# do show qos maps cos-mutation
```

- Verify the interface QoS mapping by using the **do show qos interface** command.

```
switch(config)# do show qos interface te 2/1/2
```

Configuring DSCP mappings

Consider the topics discussed below when configuring the DSCP mappings.

Configuring the DSCP trust mode

Like QoS trust mode, the Differentiated Services Code Point (DSCP) trust mode controls the user-priority mapping of incoming traffic. The user priority is based on the incoming DSCP value. When this feature is not enabled, DSCP values in the packet are ignored.

When DSCP trust is enabled, the following table shows default mapping of DSCP values to user priority.

TABLE 33 Default DSCP priority mapping

DSCP Values	User Priority
0-7	0
8-15	1
16-23	2
24-31	3
32-39	4
40-47	5
48-55	6
56-63	7

NOTE

Note the restrictions for using this feature in VCS mode under Restrictions for Layer 3 features in VCS mode on page 216.

To configure DSCP trust mode, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface.

```
switch(config)# interface tengigabitethernet 10/0/2
```

3. Set the interface mode to QoS DSCP trust.

```
switch(conf-if-te-10/0/2)# qos trust dscp
```

NOTE
To deactivate the DSCP trust mode from an interface, enter **no qos trust dscp**.

4. Return to privileged EXEC mode.

```
switch(conf-if-te-10/0/2)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Verifying DSCP trust

To verify applied DSCP trust, you can enter the following command from global configuration mode, where **tengigabitethernet 10/0/2** is the interface name.

```
switch(config)# do show qos interface tengigabitethernet 10/0/2
```

Creating a DSCP mutation map

NOTE
This feature is only supported on Brocade VDX 8770-4, VDX 8770-8, VDX 6740, VDX 6740T, and VDX 6740T-1G.

To create a DSCP mutation map and re-map the incoming DSCP value of the ingress packet to egress DSCP values, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Create the DSCP mutation map by specifying a map name. The following command uses "test" as the map name and places the system in DSCP mutation mode so that you can map to traffic classes.

```
switch(config)# qos map dscp-mutation test
```

3. Once the system is in DSCP mutation mode for the configured map (in this case dscp-mutation-test), you can map ingress DSCP values to egress DSCP values by using the **mark** command as in the following examples:

```
switch(dscp-mutation-test)# mark 1,3,5,7 to 9
switch(dscp-mutation-test)# mark 11,13,15,17 to 19
switch(dscp-mutation-test)# mark 12,14,16,18 to 20
switch(dscp-mutation-test)# mark 2,4,6,8 to 10
```

This sets the following:

- DSCP values 1, 3, 5, and 7 are set to output as DSCP number 9.
- DSCP values 11, 13, 15, and 17 are set to output as DSCP number 19.
- DSCP values 12, 14, 16, and 18 are set to output as DSCP number 20
- DSCP values 2, 4, 6, and 8 are set to output as DSCP number 10.

4. Enter the **do copy** command to save the *running-config* file to the *startup-config* file.

```
switch(config)# do copy running-config startup-config
```

Applying a DSCP mutation map to an interface

To apply a configured DSCP mutation map to an interface, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface.

```
switch(config)# interface tengigabitethernet 3/1/2
```

3. Activate or apply changes made to the DSCP mutation map to the interface. In this example "test" is the map name.

```
switch(conf-if-te-3/1/2)# qos dscp-mutation test
```

NOTE

To deactivate a map from an interface, enter **no qos dscp-mutation** *name*.

4. Specify the DSCP trust mode for incoming traffic.

```
switch(conf-if-te-2/1/2)# qos dscp-cos test
switch(conf-if-te-2/1/2)# qos dscp-traffic-class test
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-3/1/2)# end
```

6. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Verifying the DSCP mutation mapping

To verify applied DSCP maps, you can use one or both of the following options from global configuration mode.

- Verify DSCP mapping for a specific map using the **do show qos maps dscp-mutation** command and the map name.

```
switch(config)# do show qos maps dscp-mutation test
```

- Verify all DSCP mapping by using the **do show qos maps** command with the **dscp-mutation** operand only.

```
switch(config)# do show qos maps dscp-mutation
```

- Verify DSCP mutation mapping for an interface by using the **show qos interface** command and specifying the interface:

```
switch(config)# do show qos interface te 3/1/2
```

Configuring DSCP-to-CoS mappings

Consider the topics discussed below when configuring the DSCP-to-CoS mappings.

Creating a DSCP-to-CoS mutation map

You can use the incoming DSCP value of ingress packets to remap the outgoing 802.1P CoS priority values by configuring a DSCP-to-COS mutation map on the ingress interface. Use the following steps.

NOTE

The restrictions for using this feature in VCS mode are listed at Restrictions for Layer 3 features in VCS mode on page 216.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Create the DSCP-to-CoS map by specifying a map name. The following command uses "test" as the map name and places the system in dscp-cos map mode so that you can map DSCP values to CoS values.

```
switch(configure)# qos map dscp-cos test
```

3. Once the system is in dscp-cos map mode for the configured map (in this case dscp-cos-test), you can map incoming DSCP values to outgoing CoS priority values by using the **mark** command as in the following examples:

```
switch(dscp-cos-test)# mark 1,3,5,7 to 3
switch(dscp-cos-test)# mark 11,13,15,17 to 5
switch(dscp-cos-test)# mark 12,14,16,18 to 6
switch(dscp-cos-test)# mark 2,4,6,8 to 7
```

   This sets the following:

   - DSCP values 1, 3, 5, and 7 are set to output as CoS priority 3.
   - DSCP values 11, 13, 15, and 17 are set to output as CoS priority 5
   - DSCP values 12, 14, 16, and 18 are set to output as CoS priority 6
   - DSCP values 2, 4, 6, and 8 are set to output as CoS priority 7.

4. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch(config)# copy running-config startup-config
```

Applying a DSCP-to-CoS map to an interface

To apply a DSCP-to-CoS mutation map to an interface, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface.

```
switch(config)# interface tengigabitethernet 1/1/2
```

3. Apply the changes made to the DSCP-to-CoS mutation map to enable DSCP trust on the interface. In this example, "test" is the map name.

```
switch(conf-if-te-1/1/2)# qos dscp-cos test
```

   NOTE

   To deactivate a map from an interface, enter **no qos dscp-cos** *name*.

4. Apply the changes made to the DSCP-to-Traffic-Class map to enable DSCP trust on the interface. In this example, "traffic_test" is the map name.

```
switch(conf-if-te-1/1/2)# qos dscp-traffic-class traffic_test
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-1/1/2)# end
```

6. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Verifying a DSCP-to-CoS mutation map

To verify applied DSCP-to-CoS maps, you can use one or both of the following options from global configuration mode.

- Verify DSCP mapping for a specific map using the **do show qos maps dscp-cos** command and the map name.

```
switch(config)# do show qos maps dscp-cos test
```

- Verify all DSCP mapping by using the **do show qos maps** command with the **dscp-cos** operand only.

```
switch(config)# do show qos maps dscp-cos
```

- Verify DSCP-to-CoS mutation mapping for an interface by using the **show qos interface** command and specifying the interface:

```
switch(config)# do show qos interface te 1/1/2
```

Configuring traffic class mapping for flexibility

Where additional flexibility in queue selection is required, refer to Configuring traffic class mapping on page 227.

Configuring traffic class mapping

The Brocade switch supports eight unicast traffic classes to provide isolation and to control servicing for different priorities of application data. Traffic classes are numbered from 0 through 7, with higher values designating higher priorities.

NOTE
For information on user-priority mapping, refer to User-priority mapping on page 208.

The traffic class mapping stage provides some flexibility in queue selection:

- The mapping may be many-to-one, such as mapping one-byte user priority (256 values) to eight traffic classes.
- There may be a nonlinear ordering between the user priorities and traffic classes.

NOTE
The command **qos trust cos** is not applicable in VCS mode. However, the **show qos interface** command will show trusted ports if the **cos-mutation** command and the **cee** *default* command are applied.

Understanding unicast and multicast traffic defaults

The following table displays the Layer 2 default traffic **unicast** class mapping supported for a CoS-based user priority to conform to .

TABLE 34 Default user priority for unicast traffic class mapping

User priority	Traffic class
0	1
1	0
2	2
3	3
4	4
5	5
6	6
7	7

You are allowed to override these default traffic class mappings per port. Once the traffic class mapping has been resolved, it is applied consistently across any queuing incurred on the ingress and the egress ports.

The Brocade switch supports eight multicast traffic classes for isolation and to control servicing for different priorities of application data. Traffic classes are numbered from 0 through 7, with higher values designating higher priorities. The traffic class mapping stage provides some flexibility in queue selection.

The following table displays the Layer 2 default traffic **multicast** class mapping supported for a CoS-based user priority to conform to 802.1Q default mapping.

TABLE 35 Default user priority for multicast traffic class mapping

User Priority	Traffic class
0	1
1	0
2	2
3	3
4	4
5	5
6	6
7	7

Once the traffic class mapping has been resolved for inbound traffic, it is applied consistently across all queuing incurred on the ingress and egress ports.

> **NOTE**
> You can configure an interface with a nondefault DSCP-to-traffic class-map. However, configuring an interface with a nondefault CoS-to-traffic class-map is not supported.

BUM storm control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. Broadcast, unicast, and unknown multicast (BUM) storm control can prevent disruptions on Layer 2 physical ports. This feature is supported only at the interface level.

BUM storm control allows you to limit the amount of broadcast, unknown unicast, and multicast ingress traffic on a specified interface or on the entire system. All traffic received in excess of the configured rate gets discarded. You also have the option to specify whether to shut down an interface if the maximum defined rate is exceeded within a five-second sampling period. When a port is shut down, you receive a log message. You must then manually re-enable the interface by using the **no shut** command.

BUM storm control considerations and limitations

- BUM storm control must be configured on one of the following physical interfaces:
  - 1-gigabit Ethernet
  - 10-gigabit Ethernet
  - 40-gigabit Ethernet
  - 100-gigabit Ethernet
- BUM storm control and input service-policy are mutually exclusive features. Only one can be enabled at a time on a given interface.

- BUM storm control replaces the multicast rate-limit feature for Brocade VDX 6740 series, VDX 6940 series, VDX 8770-4 and VDX 8770-8, and later platforms.

Configuring BUM storm control

Refer also to BUM storm control on page 228.

To configure storm control on the 10-gigabit Ethernet interface 101/0/2, with the **broadcast** traffic type and **limit-rate** of 1000000 bps, perform the following steps:

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface for the traffic you want to control. In the following example, interface 101/0/2 is in *rbridge-id/slot/port* format:

```
switch(config)# interface tengigabitethernet 101/0/2
```

3. Issue the **storm-control ingress** command to set a traffic limit for broadcast traffic on the interface:

```
switch(conf-if-te-101/0/2)# storm-control ingress broadcast 1000000
```

4. Verify the storm control verification with the **show storm-control** command.

```
switch(conf-if-te-101/0/2)# do show storm-control
Interface  Type            rate (Mbps) conformed   violated    total
Te102/4/1  broadcast       100,000     12500000000 12500000000 25000000000
Te102/4/1  unknown-unicast 100,000     12500000000 12500000000 25000000000
Te102/4/1  multicast       100,000     12500000000 12500000000 25000000000
Te102/4/2  broadcast       100,000     12500000000 12500000000 25000000000
Te102/4/3  broadcast       100,000     12500000000 12500000000 25000000000
Te102/4/4  unknown-unicast 100,000     12500000000 12500000000 25000000000
```

NOTE

To deactivate storm control from an interface, enter **no storm-control ingress** followed by the mode (**broadcast**, **unknown-unicast**, or **multicast**) the limit (**limit-bps** or **limit-percent**), **rate**, and optionally either **monitor** or **shutdown**.

Configuring DCB QoS

Refer also to Data Center Bridging QoS on page 214.

Creating a DCB map

To create a DCB map, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Select the DCB map by using the **cee-map** command.

   The only map name allowed is "default."

```
switch(config)# cee-map default
```

3. Return to privileged EXEC mode.

```
switch(config)# exit
```

4. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Defining a DCB priority group table

To define a priority group table map, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the name of the DCB map to define by using the **cee-map** command.

> NOTE
> The only map name allowed is "default."

```
switch(config)# cee-map default
```

3. Define the DCB map for PGID 0.

```
switch(config-cee-map-default)# priority-group-table 0 weight 50 pfc on
```

4. Define the DCB map for PGID 1.

```
switch(config-cee-map-default)# priority-group-table 1 weight 50 pfc off
```

5. Return to privileged EXEC mode.

```
switch(config-cee-map-default)# end
```

6. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Defining a DCB priority-table map

To define a priority-table map, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the name of the DCB map to define by using the **cee-map** command.

```
switch(config)# cee-map default
```

3. Define the map.

```
switch(config-cee-map
)# priority-table 1 1 1 0 1 1 1 15.0
```

> NOTE
> For information about priority-table definitions, refer to the **cee-map (configuration)** command in the *Network OS Command Reference*

4. Return to privileged EXEC mode.

```
switch(config-cee-map)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Applying a DCB provisioning map to an interface

To apply a DCB provisioning map, perform the following steps from privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface. In this example, 101/0/2 is used.

```
switch(config)# interface tengigabitethernet 101/0/2
```

3. Apply the DCB map on the interface.

```
switch(conf-if-te-101/0/2)# cee default
```

NOTE
To deactivate the map on the interface, enter **no cee**.

4. Return to privileged EXEC mode.

```
switch(conf-if-te-101/0/2)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Verifying the DCB maps

To verify the CoS DCB map, use the **show cee maps default** command from privileged EXEC mode.

```
switch# show cee maps default
```

Manually enabling lossless RDMA over Ethernet

Manually enabling lossless Remote Direct Memory Access (RDMA) over Ethernet is an alternate to the default automatic method of enabling lossless Ethernet.

In Network OS, lossless 10 Gbps or 40 Gbps Ethernet is enabled automatically when an attached host signals the desire to do so through the Data Center Bridging Capability Exchange protocol (DCBX) extension mechanism

DCBX advertises this lossless request by exchanging protocol information through Type, Length, and Value attributes (DCBX TLV). This is an automatic mechanism that does not require Network OS configuration because both LLDP and DCBX are enabled by default. This is used to enable lossless Ethernet for FCoE and for some iSCSI targets which support the DCBX TLV, such as Dell EqualLogic arrays.

There are cases where lossless 10 Gbps or 40 Gbps Ethernet is desired, but where the host does not support enabling this automatically through DCBX TLV. A common example is to support Remote Direct Memory Access (RDMA) over Ethernet (also known as “RDMA over Converged Enhanced Ethernet” or “RoCE”). One example of this is to support SMB Direct in Microsoft Windows Server 2012.

Networks adapters which support RDMA over Ethernet need to have DCB Priority Flow Control (DCB PFC) enabled, and the network switches in the path also need to have DCB PFC set to the same priority in order to provide lossless Ethernet. The following are the steps needed to manually configure lossless Ethernet in Network OS (Network OS 4.0.1 or greater is required):

1. For VDX switch interfaces that are attached to hosts requiring lossless Ethernet support, disable LLDP at the 10GbE or 40GbE interface level by issuing the **lldp disable** command.
2. Also at the interface level, enable the cee map by issuing the **cee default** command.
3. Configure the host's network adapter to enable DCB PFC and to set it for PFC class 3 (priority 3). This will match the default PFC class setting in Network OS. These instructions are provided by the network adapter manufacturer.
4. It is important that the DCB PFC settings match on both the VDX switch and the host's network adapter. Ensure that the global NOS setting for the CEE map remains at its default: cee-map default

This simplifies configuration and ensures that lossless Ethernet will be automatically and correctly set up across the VCS fabric. This gives the added benefit of automatic, end-to-end lossless Ethernet throughout the VCS Ethernet fabric. When lossless Ethernet is enabled manually for PFC class 3, this will preclude support of FCoE or other protocols on PFC class 3. For this reason, do not mix FCoE configurations with manual lossless Ethernet configurations.

Configuring Brocade VCS Fabric QoS

Refer also to Brocade VCS Fabric QoS on page 216.

To configure the remapping priorities for the Brocade VCS Fabric, perform the following steps from global configuration mode.

1. Use the **cee-map** command to enter CEE map configuration mode.

```
switch(config)# cee-map default
```

2. Use the **remap lossless priority** command to set the lossless priority for Brocade VCS Fabric QoS.

   The default lossless remap priority is set to 0 (zero).

```
switch(config-cee-map-default)# remap lossless-priority priority 2
```

3. Use the **remap fabric priority** command to set the fabric priority for Brocade VCS Fabric QoS.

   The default FCoE remap fabric priority is set to 0 (zero).

```
switch(fabric-cee-map-default)# remap fabric-priority priority 2
```

4. Use the **exit** command to return to global configuration mode.

```
switch(config-cee-map)# exit
```

5. Specify the incoming Ethernet interface.

```
switch(config)# interface tengigabitethernet 22/0/1
```

6. Apply the CEE Provisioning map to the interface.

```
switch(conf-if-te-22/0/1)# cee default
```

Configuring DCB QoS

NOTE
Flow-based QoS functions only in the ingress direction.

To configure flow-based QoS functions, do the following while the switch is in global configuration mode:

1. Configure a class-map to classify traffic according to the traffic properties required for your flow-based QoS needs. Refer to Configuring a class-map on page 233.
2. Configure a policy-map to associate a policy-map with the class-map and also add the QoS action to be applied on the type of flow determined by the class-map. Refer to Configuring flow-based QoS actions using policy-map on page 234.
3. Bind the policy-map to a specific interface using the **service-policy** command, or bind the policy-map to an Rbridge ID. Refer to Binding the policy-map to an interface on page 237 and Binding flow-based QoS at the system level on page 238.

Configuring flow-based QoS classifications using a class-map

Consider the topics discussed below when configuring the flow-based QoS classifications.

Configuring a class-map

The classification map or *class-map* classifies traffic based on match criteria that you configure using the **class-map** command. If traffic matches the criteria, it belongs to the class. Currently, the only match criterion is "match access-group".

To configure a class-map, use the following steps:

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Create an access-list (either a MAC, IP, or VLAN-based ACL) to define the traffic. Refer to the Network OS Security Guide for details on creating access-lists.

```
switch(config)# mac access-list standard ACL1
switch(conf-macl-std)# permit host 0000.00aa.aa00
switch(conf-macl-std)# exit
```

3. Create a class-map by providing a class-map name. This enables class-map configuration mode.

```
switch(config)# class-map class1
```

   The name for the class-map (in this case class1) can be a character string up to 64 characters.

   NOTE
   The "default" class-map and "cee" class-map name is reserved and intended to match everything. It is always created and cannot be removed.

4. Provide match criteria for the class.

```
switch(config-classmap)# match access-group ACL1
```

5. Exit the class-map configuration mode.

```
switch(config-classmap)# exit
```

6. Return to privileged EXEC mode.

```
switch(config)# end
```

7. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

   NOTE
   Enter the **no class-map** *name* command while in global configuration mode to remove the classification map.

Configuring flow-based QoS actions using policy-map

Configure a rate-limit policy-map to associate a policy-map with the class-map and add a QoS action to be applied to the type of QoS flow defined by the class-map.

To configure a policy-map, add a classification map, and configure QoS policing parameters for the classification map, use the following steps:

1. Enter the global configuration mode.

```
switch# configure terminal
```

2. Configure a policy-map by providing a policy-map name. This enables policy-map configuration mode.

```
switch(config)# policy-map pmap1
```

3. Exit the policy-map configuration mode.

```
switch(config-policymap)# exit
```

4. Return to privileged EXEC mode.

```
switch(config)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Configuring flow-based QoS actions using policy-map

Consider the topics discussed below when configuring the flow-based QoS actions.

Configuring QoS policer action

Add color-based priority CoS mapping by configuring a policer priority-map. A policer priority-map remaps frame class of service CoS values (802.1p priority bits in VLAN tag) to conform or exceed color values when rates conform to or exceed limits set in a classification map.

The policer priority-map re-marks CoS values according to color-based green (conform), yellow (exceed), and red (violate) priorities. Creating a policer priority-map is optional. If you do not define priority mapping for a color, the map defaults to priorities of 0, 1, 2, 3, 4, 5, 6, and 7 (in other words, nothing is modified). You can configure a maximum of 32 priority-maps (one is reserved as the default), but only one map can be associated with a policer.

NOTE
You can set a priority-map when creating a policy-map by using appropriate policer attributes.

To configure a priority-map, use the following steps:

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Create a priority-map by providing a priority-map name. This enables police priority-map configuration mode.

```
switch(config)# police-priority-map pmap1
```

The name for the priority-map (in this case pmap1) can be a character string up to 64 characters.

3. Create color-based priority mapping. The following example sets the CoS for traffic that conforms to the CIR set in the policy-map.

```
switch(config-policepmap)# conform 0 1 1 2 1 2 2 1 1
```

   The following example sets the CoS for traffic that exceeds the CIR setting, but conforms to the EIR set in the policy-map.

```
switch(config-policepmap)# exceed 3 3 3 3 4 5 6 7
```

4. Return to global configuration mode with the **exit** command.

```
switch(config-policepmap)# exit
switch(config)#
```

5. Configure a policy-map by providing a policy-map name. This enables policy-map configuration mode.

```
switch(config)# policy-map pmap1
```

6. Configure a class-map in the policy-map by providing the class-map name. This enables policy class-map configuration mode. Note that the class-map name in the following example matches the name provided when you create the class-map by using the **class-map** command (refer to Configuring a class-map on page 233).

```
switch(config-policymap)# class class1
```

7. Set QoS and policing parameters for the class-map as shown in the following example. For information on all of the optional parameters for this command, refer to the *Network OS Command Reference*.

```
switch(config-policymap)# police cir 40000 cbs 5000 eir 40000 ebs 3000 set-priority pmap1 conform-
set-dscp 61 conform-set-tc 7 exceed- set-dscp 63 exceed-set-tc 3
```

   The CIR parameter is mandatory for the QoS policer. All other parameters are optional. Note that the parameter for set-priority (pmap1) includes the name for the created priority-map (refer to Configuring QoS policer action). For details on setting QoS and policing parameters, refer to Policing parameters on page 217.

8. Return to privileged EXEC mode with the **end** command.

```
switch (config-policymap)# end
```

9. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Configuring QoS mutation map actions

You can specify the mutation-map to be used on a port. This can lead to possible contradictions if there are other user-defined classes used in the same policy-map that have a set **cos action** configured. In this case-defined cos takes priority over the **mutation map**.

Perform the following task in global configuration mode.

1. Select the policy-map.

```
switch(config)# policy-map p1
```

2. Select the class.

```
switch(config-policymap)# class class-default
```

3. Specify the mutation map to be used on the port. Different kinds of mutations can be used depending on the command. For complete information, refer to *Network OS Command Reference*. The available commands are:

- dscp-mutation
- dscp-cos
- dscp-traffic-class

```
switch(config-policyclass)# map cos-mutation plsmap
```

Configuring QoS shaping action

You can specify the shaping rate per port attached to the policy-map. You can use this command to smooth out the traffic that egresses an interface. This command is allowed only for the egress direction.

NOTE
The minimum shaping speed on a Brocade VDX 6740is 200,000 Kbps.

Perform the following task in global configuration mode.

1. Select the policy-map.

```
switch(config)# policy-map p1
```

2. Select the class.

```
switch(config-policymap)# class class-default
```

3. Specify the shaping rate for the port.

```
switch(config-policyclass)# shape 30000
```

Configuring the QoS scheduling action

You can specify the scheduling attributes along with per TC shape rate. There are total of eight queues on an interface. The number of DWRR queues present depends on the SP_COUNT value. For example, if the SP_COUNT is 2, then there are two strict priority queues and six DWRR queues. This command is allowed only for the egress direction.

Perform the following task in global configuration mode.

1. Select the policy map.

```
switch(config)# policy-map p1
```

2. Select the class.

```
switch(config-policymap)# class class-default
```

3. Specify the scheduling attributes. For complete information, refer to the *Network OS Command Reference*.

```
switch(config-policyclass)# scheduler strict-priority 3 dwrr 10 10 10 10 60 TC5 35000 TC6 36000 TC7
37000
```

Configuring the sFlow profile action

You can specify the sFlow profile attached to the policy map.

This feature only functions in the the ingress direction. It can be configured both in user-defined class-maps and in the universal class-map "default". If you use the class-map "default", port-based sFlow is enabled.

Refer to Configuring an sFlow policy map and binding it to an interface on page 264.

Perform the following task in global configuration mode.

1. Select the policy-map.

```
switch(config)# policy-map p1
```

2. Select the class.

```
switch(config-policymap)# class class1
```

3. Specify the sFlow map for the port.

```
switch(config-policyclass)# map sflow mysflowmap
```

Configuring the CEE map action

The priority-mapping-table can support features provided by the Cisco Modular Quality of Service (MQC) provisioning mode to bring partial Converged Enhanced Ethernet (CEE) map content into an MQC class. MQC does not allow ingress and egress feature to be present in a same policy-map. By definition, they are two different entities and should be provisioned through two separate policy-maps. However, a CEE map provisions ingress and egress features in a single provisioning command. Because of this conflict, only the following features are inherited from a CEE map.

- Priority-Group Table.
- Priority-Mapping Table.
- Priority Flow Control Configuration.
- Lossless Priority Remapping.
- Fabric Priority Remapping.

NOTE
In Brocade switches, the CEE map scheduler configuration is global. Unless an egress scheduling policy is applied on an interface, the default scheduler is present.

Perform the following task in global configuration mode.

1. Select the policy-map.

```
switch(config)# policy-map p1
```

2. Select the class.

```
switch(config-policymap)# class cee
```

3. Attach the policy-map to the CEE map.

```
switch(config-policyclass)# priority-mapping-table import cee default
```

Configuring flow-based QoS targets

Consider the topics discussed below when configuring the flow-based QoS targets. You may apply one policy-map per interface for inbound traffic direction by using the service-policy command. Flow-based QoS functions only in the ingress direction.

Binding the policy-map to an interface

Use the **service-policy** command to associate a policy-map to an interface to apply policing parameters.

1. Enable the global configuration mode.

```
switch# configure terminal
```

2. Specify the Ethernet interface, as in the following 10-gigabit Ethernet example

```
switch(config)# interface te 1/1/2
```

3. Bind a policy-map to inress traffic on the interface. The following associates binds policymap1 to outbound traffic on the interface.

```
switch(config-if-te-1/1/2)# service-policy in policymap1
```

You can unbind the policy-map by using the **no** keyword.

```
switch(config-if-te-1/1/2)# no service-policy in
```

4. Bind a policy-map to inbound traffic on the interface. The following associates binds policymap1 to inbound traffic on the interface.

```
switch(config-if-te-1/1/2)# service-policy in policymap1
```

You can unbind the policy-map by using the **no** keyword.

```
switch(config-if-te-1/1/2)# no service-policy in
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-1/1/2)# end
```

6. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Policer binding rules

Consider the following rules when binding a policy-map to an interface:

- You can bind the same policy-map to multiple interfaces but only one policy per interface per direction is allowed.
- You cannot bind policy-maps to an interface if the policy-map has no class-map associations.

Binding flow-based QoS at the system level

Use the **qos service-policy** command to bind an existing policy-map to a single Rbridge ID or all Rbridge IDs to apply policing parameters to the interfaces in a VCS fabric.

Refer to Configuring policer functions for information on creating a service policy.

NOTE

The class-maps titled "default" and "cee" can not be bound at the system level.

1. Enable the global configuration mode.

```
switch# configure terminal
```

2. Bind the policy-map to inbound traffic.

```
switch(config)# qos service-policy in pmap1
```

3. Bind a policy-map to ingress traffic on the Rbridge ID. The Rbridge ID can be a single ID, a range of IDs, or you may use all to bind the policy-map to all Rbrdge-IDs. The following associates binds policymap1 to inbound traffic on RBridge ID 14 through 18. Only one policy per interface per direction is allowed.

```
switch(config-service-policy)# attach rbridge-id add 14-18
```

4. Return to privileged EXEC mode.

```
switch(config-service-policy)# end
```

5. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Displaying flow-based QoS configuration and operational data

Consider the topics discussed below when displaying policing settings and policy-maps.

Displaying policy-maps

In the following example, the **show policymap** command is used to display Policer policies and parameters set for the 10-gigabit Ethernet interface 4/1 inbound traffic.

```
switch(conf-if-te-5/1/33)# do show policy-map interface tengigabitethernet 5/1/33
Ingress Direction :
 Policy-Map pmap1
    Class default
      Police cir 43454
        Stats:
          Operational cir:39944 cbs:6518 eir:0 ebs:0
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
Egress Direction :
 Policy-Map pmap1
    Class default
      Police cir 43454
        Stats:
          Operational cir:39944 cbs:6518 eir:0 ebs:0
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

Entering **show policymap system rbridge-id** displays the policy-map information for the specified Rbridge ID.

```
switch(conf-if-te-2/0/11)# do show policy-map system rbridge-id 14

Ingress Direction :
 Policy-Map pmap1
    Class cmap1
      matches 0 packets
      Police cir 43454
        Stats:
          Operational cir:39944 cbs:6518 eir:0 ebs:0
          Conform Byte:0 Exceed Byte:0 Violate Byte:0
```

Entering **show policymap** without identifying an interface and specify inbound traffic displays policy-maps bound on all switch interfaces.

```
switch(conf-if-te-5/1/33)# do show policy-map
Number of policy maps : 1

Policy-Map pmap1
  Bound To: Te 5/1/33(in), Te 5/1/33(out)
            Rbridges:14,15,16,17,18

switch(conf-if-te-5/1/33)# do show policy-map detail pmap1
Policy-Map pmap1
    Class cmap1
```

```
      Police cir 43454

  Bound To: Te 5/1/33(in), Te 5/1/33(in)
            Rbridges:14,15,16,17,18
```

The following example displays the running configured policy-map by means of the **show running-config policy-map** command.

```
switch(conf-if-te-5/1/33)# do show running-config interface tengigabitethernet 5/1/33
interface TenGigabitEthernet 5/1/33
 service-policy in pmap1
 service-policy in pmap1
 fabric isl enable
 fabric trunk enable
 no shutdown
```

The following example displays the running configured service-policy by means of the **show running-config qos** command.

```
switch(conf-if-te-2/0/11)# do show running-config qos service-policy
qos service-policy in pmap1
 attach rbridge-id add 14-18
```

Displaying class-maps

The following example displays the running configured class-map name and configured match attribute by means of the **show running-config class-map** command.

```
switch(config-classmap)# do show running-config class-map
class-map cee
!
class-map class_map1
 match access-group stdacl1
!
class-map default
```

Configuring Auto QoS

Auto QoS automatically classifies traffic based on either a source or a destination IPv4 or IPv6 address. Once the traffic is identified, it is assigned to a separate priority queue. This allows a minimum bandwidth guarantee to be provided to the queue so that the identified traffic is less affected by network traffic congestion than other traffic.

> NOTE
> As this command was created primarily to benefit Network Attached Storage devices, the commands used in the following sections use the term “NAS”. However, there is no strict requirement that these nodes be actual NAS devices, as Auto QoS will prioritize the traffic for any set of specified IP addresses.

Auto QoS for NAS

There are four steps to enabling and configuring Auto QoS for NAS:

1. Enable Auto QoS.
2. Set the Auto QoS CoS value.
3. Set the Auto QoS DSCP value.
4. Specify the NAS server IP addresses.

For detailed instructions, refer to Enabling Auto QoS for NAS on page 242.

Auto QoS configuration guidelines

When configuring Auto QoS, the following configuration guidelines should be followed.

- Auto QoS is enabled and disabled globally.
- Auto QoS supports virtual fabrics.
- When Auto QoS is enabled, you can modify the CEE map subject to the following restrictions:
  - You can set the Auto QoS class of service (CoS) to any value other than “fabric” or “fcoe priority”.
  - Only one CoS can map to PGID 3.
  - PGID 2 and PGID 3 in the CEE map cannot be deleted.
  - The priority table cannot be deleted. (This sets CoS to be mapped to strict priority PGID.)
  - Strict-priorities cannot be assigned to CoS values, with the exception of strict-priority 15.0 being assigned to cos 7.
  - The Auto QoS CoS value (the PGID mapping) cannot be modified in the priority table.
- Avoid mixing L2 level multitenancy and L3 level multitenacy (VLAN and VRF), as this will cause Auto QoS statistics to not be displayed properly.
- The source and destination server IP addresses identifying NAS traffic are contained in the NAS server IP list. Refer to Specifying NAS server IP addresses for Auto QoS on page 244 and Removing NAS server IP addresses for Auto QoS on page 244 for instructions on adding and removing NAS server IP addresses.
- Enter more specific entries (those using fewer wild card values) first to have statistics display properly. For example , if you enter **nas server-ip** *10.10.0.0/16* followed by **nas server-ip** *10.10.10.0/24*, the number of matched packets will always be zero for the second entry.

Auto QoS restrictions

- Auto QoS can only be activated when the CEE map is set to the default values.
- If long distance ISL is configured, you cannot enable Auto QoS.
- Different NAS traffic types are not distinguished, and all NAS traffic types are treated equally. This means all NAS traffic, whatever the type, share a common bandwidth guarantee. Individual traffic types do not get separate bandwidth guarantees.
- In order to provide a minimum bandwidth guarantee for NAS traffic across switches in the fabric, when Auto QoS is enabled for NAS, the CEE map reserves 20% of the available bandwidth for NAS traffic. After enabling Auto-nas, NAS bandwidth can be modified by modifying CEE-MAP.
- Port-level configuration has higher precedence than the NAS configuration. This means that any interface-specific configuration will override the global configuration.
- The *conform* and *exceed* traffic class values are not set for the NAS traffic class when Auto QoS for NAS is enabled.
- If all QoS Ternary Content-Addressable Memory (TCAM) space is used up already, enabling Auto QoS will not have any impact on NAS traffic.
- The ACL byte counter is not supported on Brocade VDX 6740 and VDX 6940 series platforms.
- PFC is always turned OFF for the NAS classified traffic and cannot be made lossless.
- Auto QoS traffic classification is always done in the ingress RBridge. Once the traffic is classified, all other nodes in the cluster automatically provision the default bandwidth guarantee of 20% through their CEE map.
- Automatic Migration of Port Profiles (AMPP) and Auto-NAS cannot be active on the same switch. This means that if the CEE map is part of AMPP port profile, Auto QoS cannot be enabled, and if Auto QoS is enabled and the CEE map is then associated to AMPP port profile, Auto QoS cannot be disabled. In order to disable Auto QoS, you must:
  1. Disassociate the CEE map from the port profile.
  2. Disable Auto QoS.
  3. Associate the CEE map with the AMPP port profile.

Auto QoS and CEE maps

In order to provide minimum bandwidth guarantee for NAS traffic across switches in the fabric, the default CEE map has to be changed to accommodate NAS traffic. By default, the bandwidth division in the CEE map is 40% for FCoE traffic and 60% for LAN traffic. As NAS traffic has to co-exist with FCoE traffic, some of the LAN traffic bandwidth allocation is given over to NAS traffic. When Auto QoS is enabled, the default bandwidth allocations are 40% for FCoE traffic, 20% for NAS traffic and 40% for LAN traffic. The purpose of using a CEE map is to pass along the scheduler weights to ISLs in the fabric so that they will treat the NAS traffic accordingly.

When Auto QoS is enabled, the modified CEE map will be similar to the following:

```
switch# show cee maps

CEE Map 'default'
   Precedence: 1
   Remap Fabric-Priority to Priority 0
   Remap Lossless-Priority to Priority 0
   Priority Group Table
     1:  Weight 40, PFC Enabled, BW% 40
     2:  Weight 40, PFC Disabled, BW% 40
     3:  Weight 20, PFC Disabled, BW% 20
     15.0: PFC Disabled
     15.1: PFC Disabled
     15.2: PFC Disabled
     15.3: PFC Disabled
     15.4: PFC Disabled
     15.5: PFC Disabled
     15.6: PFC Disabled
     15.7: PFC Disabled

     Priority Table
     CoS:     0    1    2    3    4    5    6   7
     ------------------------------------------------
     PGID:    2    2    3    1    2    2    2   15.0
```

In the example above a PGID with an ID of "3" is defined with a bandwidth allocation of 20 which is then mapped to a user-configured CoS value. If the CoS value is not configured, the PGID value is mapped to the default QoS CoS value (2) in the priority table. This shows that when this CEE map is applied to an interface, the CoS will be allotted 20% of the overall bandwidth.

Enabling Auto QoS for NAS

Auto QoS (Quality of Service) for NAS creates a minimum bandwidth guarantee for Network Attached Storage traffic. Auto QoS for NAS is disabled by default; you must enable Auto QoS to allow tagged NAS packets to have the correct service levels.

All steps must be performed in route-map configuration mode.

The **cee-map** priority group and priority-map settings must be their default values.

Enabling Auto QoS for NAS:

- Changes the CoS value of tagged NAS packets to 2
- Reduces the weight of PGID 2 from 60 to 40
- Creates a new PGID 3 with a weight of 20
- Modifies the priority table to include PGID 3 for the user-configured NAS CoS, or the default NAS CoS if the CoS has not been otherwise modified

Use the following procedure to enable Auto QoS for NAS traffic.

1. Enable Auto QoS for all NAS traffic by entering **nas auto-qos**.

   ```
   switch(config)# nas auto-qos
   ```

2. Set the Class of Service for all NAS traffic by entering: **set cos** *cos_value*.

The CoS value affects how Auto QoS operated by specifying the User-Priority field value and traffic-class value in the VLAN packet header. If you do not specify a CoS value, the NAS CoS value is set to the default of 2.

This example sets the CoS value to 3:

```
switch(config)# set cos 3
```

3. Set the DSCP value for all NAS traffic by entering **set dscp** *dscp_value*.

   The Differentiated Services Code Point (DSCP) value affects how Auto QoS operates by specifying the priority value for Network Attached Storage traffic on IP networks. If you do not specify a DSCP value, the DSCP value is set to the default of 0. Higher numbers provide a higher level of priority.

   The following example sets the DSCP value to 56:

```
switch(config)# set dscp 56
```

4. Identify the IPv4 network addresses (either origination or destination) used by the NAS devices by entering **nas server-ip** followed by the IP address including the mask and then one of the following:
   - **vlan** *vlan_ID*
   - **vrf** *vrf_Name*

     NOTE
     Associating both a VRF and a VLAN value to the same server IP address is strongly discouraged, as this will cause errors in reporting NAS statistics.
5. Press **Enter** after you add each address entry.

   The following example adds two addresses, one using a VLAN mask, and the other a VRF mask.

```
switch(config)# nas server-ip 10.192.100.100/32 vlan 100
switch(config)# nas server-ip 10.192.100.101/32 vrf bruce
```

Disabling Auto QoS for NAS

Disabling Auto QoS (Quality of Service) for NAS removes the minimum bandwidth guarantee for Network Attached Storage traffic.

Disabling Auto QoS for NAS:

- Disables Auto QoS functionality for NAS
- Restores the default bandwidth settings if you have not made any changes to the CEE map after enabling Auto QoS. If you have made changes, the portion of the bandwidth you assigned to NAS traffic will revert to the LAN bandwidth.
- Replaces PGID 3 with PGID 2 in the priority table
- Deletes PGID 3
- Increases the weight of PGID 2 by the weight of PGID 3, so that the weight of PGID 2 equals the weight of PGID 2 plus the weight of PGID 3

Use the following procedure to disable Auto QoS for NAS traffic and restore the default CoS and DSCP values.

1. Enter **no nas auto-qos** in configuration mode.

```
switch(config)# no nas auto-qos
```

2. Enter **no set cos** in route-map configuration mode to disable CoS for NAS and restore the default value.

```
switch(config)# no set cos
```

3. Enter **no set dscp** in route map configuration mode to disable DSCP for NAS and restore the default value.

```
switch(config)# no set dscp
```

Displaying Auto-NAS configurations

Network OS allows you to display Network Attached Storage server configurations for all NAS servers.

Use the following command to display the Auto-NAS configuration for a switch.

Enter **show system internal nas**.

The following example shows a typical output of this command, showing that Auto-NAS is enabled on two IP address (one using VLAN, and one using VRF), that it has the values of CoS 2 and DSCP 0, and a Traffic Class of 5.

```
switch# show system internal nas
Auto-NAS Enabled
Cos 2
Dscp 0
Traffic Class 5
NAS server-ip 10.192.100.100/32 vlan 100
NAS server-ip 10.192.100.101/32 vrf broceliande
```

Specifying NAS server IP addresses for Auto QoS

To enable Auto QoS for NAS, you must identify the IPv4 network addresses used by the NAS devices and tell the platform to tag all packets with this network address as NAS traffic so that they have the correct priority.

The **nas server-ip** command can accept a single IPv4 address (10.192.100.100/32), or an entire sub-net (10.192.100.0/24 vlan 100) as input. You can specify either a Virtual Routing and Forwarding (VRF) ID, or a VLAN ID. If no ID identifier is specified, the default VRF is assumed. When a subnet is specified, all the servers in the sub-net will have Auto QoS enabled for NAS. Subnet masks are not supported (for example: 10.192.100.0 255.255.255.0 is not a supported address).

NOTE
IPv6 addressing is also supported for Auto QoS for NAS.

To add a NAS address to the NAS server IP list:

1. In configuration mode, enter **nas server-ip** followed by the IP address with mask and then one of the following:
   - **vlan** *vlan_ID*
   - **vrf** *vrf_Name*
2. Press **Enter** after you add each address entry.

The following example adds two addresses, one using a VLAN mask, and the other a VRF mask.

```
switch(config)# nas server-ip 10.192.100.100/32 vlan 100
switch(config)# nas server-ip 10.192.100.101/32 vrf broceliande
```

Removing NAS server IP addresses for Auto QoS

Removing NAS server IP addresses sets the Auto QoS values for those addresses back to their defaults.

The **no nas server-ip** command can accept a single IPv4 address (10.192.100.100/32), or an entire sub-net (10.192.100.0/24 vlan 100) as input. When a subnet is specified, all the servers in the sub-net will have Auto QoS for NAS removed. Subnet masks are not supported (for example: 10.192.100.0 255.255.255.0 is not a supported address).

To remove an address from the NAS server IP list:

1. In config mode, enter **no nas server-ip** followed by the IPv4 address including the mask and then one of the following:
   - **vlan** *vlan_ID*
   - **vrf** *vrf_Name*
2. Press **Enter** after you add each individual address entry.

The following example removes two addresses, one using a VLAN mask, and the other a VRF mask.

```
switch(config)# no nas server-ip 10.192.100.100/32 vlan 100
switch(config)# no nas server-ip 10.192.100.101/32 vrf broceliande
```

Displaying NAS server IP addresses

How to display the IP addresses for network-attached storage (NAS) servers.

You must be in config mode to run this command.

Use the following command to display all the configured NAS server IPv4 addresses. IPv6 addresses are not supported:

show running-config nas server-ip.

The following example shows the IP addresses of the active NAS servers.

```
switch(config)# show running-config nas server-ip
nas server-ip 10.192.100.100/32 vlan 100
nas server-ip 10.192.100.101/32 vrf broceliande
```

Displaying NAS server statistics

Network OS allows you to display Network Attached Storage (NAS) server statistics for a single server or for all servers.

The **show nas statistics all** command displays the number of incoming IP packets which are NAS-classified for every RBridge in the cluster. It does not show any egressing NAS-classified packets.

Traffic matching the NAS Auto QoS server IP rule works in the same way as matching sequence entries in user-configured ACLs. If the first entry is hit, it will not proceed with any more entry checks.

The order of Auto QoS server IP addresses in the system is not necessarily (or always) same as the order of server IPs programmed in the hardware.

1. To display all NAS server statistics, enter **show nas statistics all**.

```
switch# show nas statistics all
RBridge 1
-----------
nas server-ip 10.1.1.1/32  vrf default-vrf
matches 0 packets 0 bytes

RBridge 2
-----------
nas server-ip 10.1.1.1/32 vrf default-vrf
matches 0 packets 0 bytes
```

2. To display NAS server statistics for a single server, enter: **show nas statistics server-ip** followed by the IPv4 address and the appropriate VLAN or VRF mask and RBridge ID.

   This command can accept a single IPv4 address (10.192.100.100/32), or an entire sub-net (10.192.100.0/24 vlan 100) as input. When a subnet is specified, all the servers in the sub-net with Auto-NAS QoS will be shown. Subnet masks are not supported (for example: 10.192.100.0 255.255.255.0 is not a supported address). The first following example shows the statistics for all ports using RBridge ID 1; the second example shows the statistics for 10.1.1.0/24 vrf brad.

```
switch# show nas statistics all rbridge-id 1
RBridge 1
```

```
-----------
nas server-ip 10.1.1.1/32  vrf default-vrf
matches 0 packets 0 bytes

switch# show nas statistics server-ip 10.1.1.0/24 vrf brad
nas server-ip 10.1.1.0/24 vrf brad
matches 2000000 packets 40000000 bytes
```

Clearing NAS server statistics

You can use this command to clear the statistics for a single IPv4 address (10.192.100.100/32), or an entire sub-net (10.192.100.0/24 vlan 100). When a subnet is specified, the NAS statistics for all the servers in the sub-net will be cleared. Subnet masks are not supported (for example: 10.192.100.0 255.255.255.0 is not a supported address).

1. To clear NAS server statistics for a single server, enter: **show nas statistics server-ip** followed by the IPv4 address and the appropriate VLAN or VRF mask and RBridge ID.

   The following example shows clearing the NAS statistics for "10.1.1.0/24 vrf brad" and then the effect of this clearing.

   ```
   switch# clear nas statistics server-ip 10.1.1.0/24 vrf brad
   switch# show nas statistics server-ip 10.1.1.0/24 vrf brad
   nas server-ip 10.1.1.0/24 vrf brad
   matches 0000000 packets 00000000 bytes
   ```

2. To clear all NAS server statistics, enter **clear nas statistics** *all*.

IGMP

IGMP overview

Internet Group Management Protocol (IGMP) is a communications protocol that allows hosts and routers to establish group memberships, and is an integral part of IP multicast. This chapter does not address all aspects of IGMP, but rather focuses on the snooping mechanism, as described below. For additional commands that support basic IGMP configuration, refer to Using additional IGMP commands on page 253.

IGMP snooping overview

The forwarding of multicast control packets and data through a Layer 2 switch configured with VLANs is most easily achieved by the Layer 2 forwarding of received multicast packets on all the member ports of the VLAN interfaces. However, this simple approach is not bandwidth efficient, because only a subset of member ports may be connected to devices interested in receiving those multicast packets. In a worst-case scenario, the data would get forwarded to all port members of a VLAN with a large number of member ports (for example, all 24 ports), even if only a single VLAN member is interested in receiving the data. Such scenarios can lead to loss of throughput for a switch that gets hit by a high rate of multicast data traffic.

Internet Group Management Protocol (IGMP) snooping is a mechanism by which a Layer 2 switch can effectively address this issue of inefficient multicast forwarding to VLAN port members. Snooping involves "learning" forwarding states for multicast data traffic on VLAN port members from the IGMP control (join/leave) packets received on them. The Layer 2 switch also provides for a way to configure forwarding states statically through the CLI.

Multicast routing and IGMP snooping

Multicast routers use IGMP snooping to learn which groups have members on each of their attached physical networks. A multicast router keeps a list of multicast group memberships for each attached network, and a timer for each membership.

NOTE

"Multicast group memberships" means that at least one member of a multicast group on a given attached network is available.

There are two ways that hosts join multicast routing groups:

- By sending an unsolicited IGMP join request.
- By sending an IGMP join request as a response to a general query from a multicast router.

In response to the request, the switch creates an entry in its Layer 2 forwarding table for that VLAN. When other hosts send join requests for the same multicast, the switch adds them to the existing table entry. Only one entry is created per VLAN in the Layer 2 forwarding table for each multicast group.

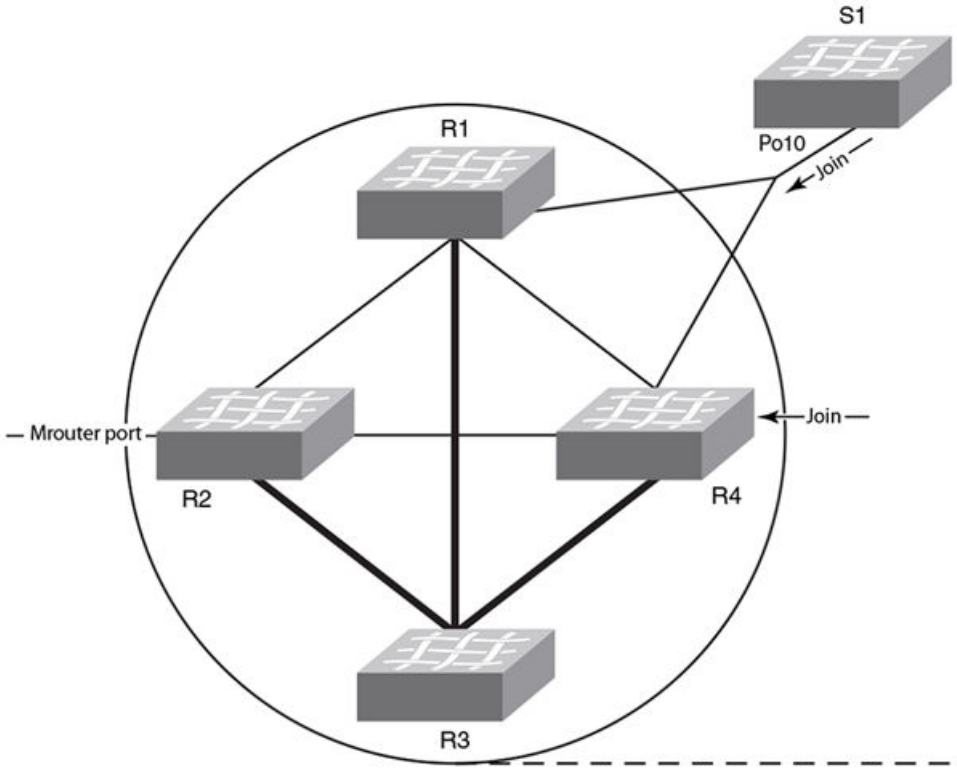
vLAG and LAG primary port with IGMP snooping

The current data center Ethernet (DCE) implementation of vLAGs and LAGs uses the concept of a so-called primary port. One of the member ports of the vLAG and LAG is selected to be the primary port, and all multicast traffic egressing from the LAG or vLAG is sent on the primary port. Thus, normal hash-based forwarding is not performed for multicast traffic, whether it is control traffic or data. Now,

consider the case where RBridge R1 receives an IGMP join request for group G1 on Po10, shown in the figure below. This causes Po10 to be added to the list of IGMP receivers for group G1. Now, assume that the primary port of the vLAG is the link connecting R4 and S1. Therefore, any multicast traffic received by the cluster for group G1 egresses on vLAG Po10 from R4 and not from R1, even though the original join was received on R1.

If the primary port for the vLAG changes, such as if the link between R4 and S1 in the figure below went down, then multicast traffic would egress out of the new primary port on the vLAG. In the above case, the new primary port would be the link connecting R1 and S1.

FIGURE 38 IGMP snooping in Brocade VCS Fabric mode



PIM multicast router presence detection

The PIM hello-based multicast router presence detection feature scans the network traffic for incoming PIM hellos. This is enabled by default.

When a PIM hello is detected, that port is marked for the presence of a multicast router and the information is saved. This prevents unnecessary flooding if the PIM designated router (DR) goes offline, as IGMP reports are forwarded to the multicast routers and not only the snooping-enabled router.

IGMP snooping scalability

Here are the scalability limits of IGMP snooping feature in various modes of switch operation for Network OS 4.1.0 and later. The table explains the various metrics involved in describing the scalability limits.

IGMP Metric	Description
Maximum number of IGMP groups supported	This metric is based on the available hardware resources, such as multicast group ID (MGID), configuration replay, and Ethernet Name Server (eNS) distribution bandwidth.
Maximum number of VLANs supported with IGMP snooping configuration	This metric is limited by the general-query packet-generation capacity of IGMP software processes running on the switch, as well as by eNS distribution bandwidth.
Maximum IGMP packet-processing rate per switch	The scalability number described by this metric suggests the upper limit on the number of packets that can be processed by IGMP software processes running on the switch. If the packets are incoming from multiple ports/VLANs, the same processing bandwidth is shared.
Maximum IGMP packet-processing rate per Brocade VCS Fabric cluster	This metric specifies the upper limit on the maximum rate of IGMP packets incoming to a logical Brocade VCS Fabric switch. It is limited by the eNS distribution bandwidth and the number of nodes in the Brocade VCS Fabric cluster.

IGMP snooping in Brocade VCS Fabric cluster mode

When supporting a flat Layer 2 network in a data center, VDX switches can be connected in any order to form a cluster. The number of nodes involved in a cluster ranges from four nodes to 24 nodes. Metrics are detailed in the following tables.

TABLE 36 IGMP snooping: four-node cluster metrics

Metric	Limit	Comments
Maximum number of IGMP groups supported	6000	Join requests are sent on four ports of the same switch.
Maximum number of VLANs supported with IGMP configuration	512	
Maximum IGMP packet-processing rate per switch	512 packets/sec	
Maximum IGMP packet-processing rate per Brocade VCS Fabric cluster	512 packets/sec	

TABLE 37 IGMP snooping: 24-node cluster metrics

Metric	Limit	Comments
Maximum number of IGMP groups supported	6000	Join requests are sent on four ports of the same switch.
Maximum number of VLANs supported with IGMP configuration	512	
Maximum IGMP packet-processing rate per switch	512 packets/sec	
Maximum IGMP packet processing rate per Brocade VCS Fabric cluster	512 packets/sec	

TABLE 38 IGMP snooping: Brocade VDX 8770-4 and VDX 8770-8 cluster metrics

Metric	Limit	Comments
Maximum number of IGMP groups supported	6000	Join requests are sent on four ports of the same switch.
Maximum number of VLANs supported with IGMP configuration	512	
Maximum IGMP packet processing rate per switch	512 packet/second	
Maximum IGMP packet processing rate per Brocade VCS Fabric cluster	512 packet/second	

TABLE 39 IGMP snooping: IP multicast metrics

Metric	Limit	Comments
Number of Layer 3 forwarding entries	256	
Number of IGMP snooping forwarding entries	6000	
Number of multicast flows	10000	

TABLE 39 IGMP snooping: IP multicast metrics (continued)

Metric	Limit	Comments
PIM interfaces supported	32	
IGMP interfaces supported	32	
IGMP snooping interfaces supported	512	
Learning rate for PIM-SM	32 flows/second	
Learning rate for IGMP snooping	512 groups/second	

Configuring IGMP snooping

By default, IGMP snooping is globally disabled on all VLAN interfaces. Refer to the *Network OS Command Reference* for complete information about the commands in this section.

IGMP snooping configuration considerations

The following configuration considerations apply to IGMP snooping beginning Network OS release 7.0.0.

- You must remove the IGMP snooping static mrouter configuration from all VLANs before upgrading or downgrading from or to the NOS 6.0.2x release.
- IGMP configuration is not supported on VEs. IGMP configuration is supported only on VLAN, router ports and port-channels.
- IGMP configurations supported under router ports are supported under port-channels.
- IGMP snooping querier and static mrouter can co-exist on a VLAN interface.
- IGMP control packets only with TTL value 1 will be processed.
- Querier and m-router configurations can co-exist.
- You must enable snooping at the global and VLAN levels. Enabling snooping at a global level does not enable snooping on any of the VLANs, however disabling snooping at the global level will disable snooping on all VLANs. This behavior is not supported on clusters where some nodes in the cluster are running NOS firmware prior to 7.0.0 and others in the cluster are running the 7.0.0 firmware.
- When upgrading from versions prior to NOS 7.0.0 to NOS 7.0.0, you must ensure the following post the upgrade procedure:
  1. Snooping is enabled at the global and VLAN level.
  2. Any IGMP configurations under VE should be moved to the VLAN level, as IGMP configuration under VE is not supported.

IGMP snooping upgrade and downgrade considerations

The following table lists the IGMP snooping upgrade and downgrade considerations on VLANs across Network OS 6.0.1 and Network OS 7.0.0.

TABLE 40 IGMP snooping upgrade and downgrade considerations

Network OS 6.0.1	Network OS 7.0.0
Fewer than 512 VLANs are configured with global snooping enabled.	**Logical Chassis mode:** IGMP snooping configuration will be present at the global and VLAN levels after upgrade. **Fabric Cluster mode:** If copying the running-config file to the startup-config file is performed before the upgrade, the IGMP snooping configuration will be present at the global and VLAN levels after upgrade. If copying the running-config to file is performed before the upgrade, you must enable snooping on VLANs after file replay.
Fewer than 512 snooping-enabled VLANs configured. Global	**Logical chassis mode:** Global snooping is automatically enabled after upgrade.

TABLE 40 IGMP snooping upgrade and downgrade considerations (continued)

Network OS 6.0.1	Network OS 7.0.0
snooping is not enabled. Only VLAN-level snooping is enabled.	**Fabric Cluster mode:** If copying the running-config file to the startup-config file is performed before the upgrade, global snooping is automatically enabled after upgrade. If copy running-config to file is performed before the upgrade, you must manually enable global snooping, as the database is not persistent.
More than 512 snooping-enabled VLANs, irrespective of whether global snooping is enabled or not.	**Logical Chassis mode:** If the number of snooping-enabled VLANs is more than 512 in Network OS 6.0.x , snooping will be disabled on all VLANs. • If global snooping is enabled and more than 512 VLANs are configured, after the upgrade, you must enable snooping on the VLANs. • If global snooping is not enabled, and if there are more than 512 snooping-enabled VLANs, after upgrading, you must enable VLAN-level and global-level IGMP snooping. This is because only 512 snooping-enabled VLANs are supported across the cluster. If this number was more than 512 in Network OS 6.0.x, snooping on VLANs will be disabled during upgrade. **Fabric Cluster mode:** If the running-config is copied to startup config before the upgrade, and there are more than 512 snooping-enabled VLANs (this can occur when global snooping is enabled) in Network OS 6.0.x, snooping will be disabled on all VLANs. If global snooping is enabled and there are more than 512 VLANs configured, after upgrading you must enable snooping on the VLANs. NOTE If global snooping is enabled and the running-config is copied to a file (no startup-config present) before upgrade, you must enable snooping on VLANs irrespective of the number of VLANs, after upgrade and file replay.
Downgrading from 7.0.0 to 6.0.x	Downgrade is not allowed if IGMP snooping is enabled either at the global or VLAN levels. The following messages display when the downgrade command is executed: **Only global snooping enabled** **Message:** - Downgrade is not allowed because global IGMP snooping is enabled. Please disable global IGMP snooping. **Only VLAN level snooping enabled** **Message:** - Downgrade is not allowed because IGMP snooping is enabled on one or more VLANs. Please disable IGMP snooping on VLANs. **Global and VLAN-level snooping enabled** **Message:** - Downgrade is not allowed because IGMP snooping is enabled globally and on one or more VLANs. Please disable global IGMP snooping and VLAN-level IGMP snooping.

Enabling IGMP snooping

Use the following procedure to enable IGMP snooping on a Data Center Bridging (DCB)/Fibre Channel over Ethernet (FCoE) switch.

You must enable snooping at the global and VLAN levels. Enabling snooping at a global level does not enable snooping on any of the VLANs, however disabling snooping at the global level will disable snooping on all VLANs. This behavior is not supported on clusters where few nodes in the cluster are running NOS firmware prior to 7.0.0 and others in the cluster are running the 7.0.0 firmware.

NOTE
From NOS release 7.0.0, IGMP configurations are supported only on VLAN, LAG, vLAG, and physical ports.

1. Enter the **configure terminal** command to access global configuration mode.

```
switch# configure terminal
```

2. Enter the **ip igmp snooping enable** command to enable IGMP snooping at the global level.

```
switch(config)# ip igmp snooping enable
```

3. Enter the **interface** command to select the VLAN interface number.

```
switch(config)# interface vlan 10
```

4. Optional: Activate the default IGMP snooping querier functionality for the VLAN.

```
switch(config-Vlan-10)# ip igmp snooping enable
```

Configuring the IGMP snooping querier

If your multicast traffic is not routed because Protocol-Independent Multicast (PIM) and IGMP are not configured, use the IGMP snooping querier in a VLAN.

The IGMP snooping querier sends out IGMP queries to trigger IGMP responses from switches that are to receive IP multicast traffic. The IGMP snooping querier listens for these responses to map the appropriate forwarding addresses.

Beginning with Network OS 7.0.0, IGMP snooping queries go out with source IP addresses as 0.0.0.0 and in a VCS cluster. The RBridge with the lowest RBridge ID gets elected as the querier.

> NOTE
> Snooping querier is suspended if Layer 3 IGMP is enabled on any of the cluster nodes.

Use the following procedure to configure the IGMP snooping querier.

1. Enter the **configure terminal** command to access global configuration mode.

```
switch# configure terminal
```

2. Enter the **interface** command to select the VLAN interface number.

```
switch(config)# interface vlan 25
```

3. Set the IGMP query interval for the VLAN.

```
switch(config-Vlan-25)# ip igmp query-interval 125
```

   The valid range is 1 through 18000 seconds. The default is 125 seconds.

4. Set the last member query interval.

```
switch(config-Vlan-25)# ip igmp last-member-query-interval 1000
```

   The valid range is 1000 through 25500 milliseconds. The default is 1000 milliseconds.

5. Set the last member query count.

```
switch(config-Vlan-25)# ip igmp last-member-query-count 3
```

   The valid range is 2 through 10. The default is 2.

6. Set the startup query count.

```
switch(config-Vlan-25)# ip igmp startup-query-count 3
```

   The valid range is 1 through 10. The default is 1.

7. Set the startup query interval.

```
switch(config-Vlan-25)# ip igmp startup-query-interval 200
```

   The valid range is 1 through 450 seconds. The default is 1 second.

8. Set the Maximum Response Time (MRT).

```
switch(config-Vlan-25)# ip igmp query-max-response-time 10
```

   The valid range is 1 through 25 seconds. The default is 10 seconds.

9. Set the snooping robustness variable.

```
switch(config-Vlan-25)# ip igmp robustness-variable 5
```

   The valid range is 2 through 10. The default is 2.

10. Activate the IGMP snooping querier functionality for the VLAN.

```
switch(config-Vlan-25)# ip igmp snooping querier enable
```

NOTE
IGMP snooping querier and static m-Router can be configured together on a VLAN interface.

Monitoring IGMP snooping

Monitoring the performance of your IGMP traffic allows you to diagnose any potential issues on your switch. This helps you utilize bandwidth more efficiently by setting the switch to forward IP multicast traffic only to connected hosts that request multicast traffic.

Refer to the *Network OS Command Reference* for complete information about the commands in this section.

Use the following procedure to monitor IGMP snooping on a DCB/FCoE switch.

1. Enter the **configure terminal** command to access global configuration mode.
2. Enter the **show ip igmp groups** command to display all information on IGMP multicast groups for the switch.

   Use this command to display the IGMP database, including configured entries for either all groups on all interfaces, all groups on specific interfaces, or specific groups on specific interfaces.

```
switch# show ip igmp groups
```

3. Use the **show ip igmp statistics** command to display the IGMP statistics for a VLAN or interface.

```
switch# show ip igmp snooping statistics interface vlan 10
```

4. Use the **show ip igmp snooping mrouter** command to display mrouter port-related information for all VLANs, or a specific VLAN.

```
switch# show ip igmp snooping mrouter
```

   or

```
switch# show ip igmp snooping mrouter interface vlan 10
```

5. When you have reviewed the IGMP statistics for the switch, refer to Configuring IGMP snooping on page 250 or Configuring the IGMP snooping querier on page 252 to make any needed corrections.

NOTE
Refer to the *Network OS Command Reference* for additional information on IGMP CLI commands.

Using additional IGMP commands

The following commands provide additional support for basic IGMP functionality. For details, refer to the *Network OS Command Reference*.

Command	Description
ip igmp immediate-leave	Removes a group from the multicast database. Use this command to treat the interface as if it had one multicast client, so that a receipt of a Leave Group request on the interface causes the group to be immediately removed from the multicast database. This command is available for router ports and Layer 3 port channels.
ip igmp snooping fast-leave	Removes a group from the multicast database. This command is available for VLANs.
ip igmp snooping restrict-unknown-multicast	Deactivates or reactivates the flooding of unregistered multicast data traffic on IPv4 IGMP snooping-enabled VLANs. This command functions only with an IPv4 or IPv6 multicast hardware profile.
ipv6 mld snooping restrict-unknown-multicast	Deactivates or reactivates the flooding of unregistered multicast data traffic on IPv6 MLDv1 IGMP snooping-enabled VLANs.
ip igmp static-group	Configures the static group membership entries for a specific interface.

802.1x Port Authentication

802.1x protocol overview

The 802.1x protocol defines a port-based authentication algorithm involving network data communication between client-based supplicant software, an authentication database on a server, and the authenticator device. In this situation the authenticator device is the Brocade VDX hardware.

As the authenticator, the Brocade VDX hardware prevents unauthorized network access. Upon detection of the new supplicant, the Brocade VDX hardware enables the port and marks it "unauthorized." In this state, only 802.1x traffic is allowed. All other traffic (for example, DHCP and HTTP) is blocked. The Brocade VDX hardware transmits an Extensible Authentication Protocol (EAP) Request to the supplicant, which responds with the EAP Response packet. The Brocade VDX hardware then forwards the EAP Response packet to the RADIUS authentication server. If the credentials are validated by the RADIUS server database, the supplicant may access the protected network resources.

When the supplicant logs off, it sends an EAP Logoff message to the Brocade VDX hardware, which then sets the port back to the "unauthorized" state.

NOTE
802.1x port authentication is not supported by LAG (Link Aggregation Group) or interfaces that participate in a LAG.

NOTE
The EAP-MD5, EAP-TLS, EAP-TTLS and PEAP-v0 protocols are supported by the RADIUS server and are transparent to the authenticator switch.

Configuring 802.1x authentication

The tasks in this section describe the common 802.1x operations that you will need to perform. For a complete description of all the available 802.1x CLI commands for the Brocade VDX hardware, refer to the *Network OS Command Reference*.

Understanding 802.1x configuration guidelines and restrictions

When configuring 802.1x, be aware of this 802.1x configuration guideline and restriction: If you globally disable 802.1x, then all interface ports with 802.1x authentication enabled automatically switch to force-authorized port-control mode.

Configuring authentication

The **radius-server** command attempts to connect to the first RADIUS server. If the RADIUS server is not reachable, the next RADIUS server is contacted. However, if the RADIUS server is contacted and the authentication fails, the authentication process does not check for the next server in the sequence.

Perform the following steps to configure authentication.

1. Enter the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **radius-server** command to add RADIUS to the switch as the authentication server. This command can be repeated for additional servers. However, this command moves the new RADIUS server to the top of the access list.

```
switch(config)# radius-server host 10.0.0.5
```

3. Enable 802.1x authentication globally

```
switch(config)# dot1x enable
```

4. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

5. Use the **dot1x authentication** command to enable 802.1x authentication.

```
switch(conf-if-te-5/1/12)# dot1x authentication
```

6. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

7. Enter the **copy** command to save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Configuring interface-specific administrative features for 802.1x

It is essential to configure the 802.1x port authentication protocol globally on the Brocade VDX hardware, and then enable 802.1x and make customized changes for each interface port. Because 802.1x is enabled and configured in Configuring 802.1x authentication on page 255, use the administrative tasks in this section to make any necessary customizations to specific interface port settings.

Enabling an 802.1x readiness check

The 802.1x readiness check monitors 802.1x activity on all the switch ports and displays information about the devices connected to the ports that support 802.1x. You can use this feature to determine if the devices connected to the switch ports are 802.1x-capable.

The 802.1x readiness check is allowed on all ports that can be configured for 802.1x. The readiness check is not available on a port that is configured by the **dot1x force-unauthorized** command.

When you configure the **dot1x test eapol-capable** command on an 802.1x-enabled port, and the link comes up, the port queries the connected client about its 802.1x capability. When the client responds with a notification packet, it is 802.1x-capable. A RASlog message is generated if the client responds within the timeout period. If the client does not respond to the query, the client is not 802.1x-capable, and a syslog message is generated saying the client is not EAPOL-capable.

Follow these guidelines to enable the readiness check on the switch:

- The readiness check is typically used before 802.1x is enabled on the switch.
- 802.1x authentication cannot be initiated while the 802.1x readiness test is in progress.
- The 802.1x readiness test cannot be initiated while 802.1x authentication is active.
- 802.1x readiness can be checked on a per-interface basis. Readiness check for all interfaces at once is not supported.
- The 802.1x test timeout is shown in **show dot1x** command.

This example shows how to enable a readiness check on a switch to query a port. It also shows the response received from the queried port verifying that the device connected to it is 802.1x-capable:

```
switch# dot1x test eapol-capable interface gigabitethernet 5/0/13
DOT1X_PORT_EAPOL_CAPABLE:DOT1X: MAC 00-01-02-4b-f1-a3 on gigabitethernet5/0/13 is EAPOL capable.
```

Configuring 802.1x port authentication on specific interface ports

To configure 802.1x port authentication on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you wish to modify.

1. Enter the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. Use the **dot1x authentication** command to enable 802.1x authentication.

```
switch(conf-if-te-5/1/12)# dot1x authentication
```

4. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

5. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Configuring 802.1x timeouts on specific interface ports

NOTE

While you are free to modify the timeout values, Brocade recommends that you leave all timeouts set to their defaults.

To configure 802.1x timeout attributes on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you wish to modify.

1. Enter the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. Configure the **dot1x timeout** interval.

```
switch(conf-if-te-5/1/12)# dot1x timeout supp-timeout 40
```

4. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

5. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Configuring 802.1x port reauthentication on specific interface ports

To configure 802.1x port reauthentication on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you want to modify.

1. Enter the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. Use the **dot1x authentication** command to enable 802.1x authentication for the interface port.

```
switch(conf-if-te-5/1/12)# dot1x authentication
```

4. Configure reauthentication for the interface port.

```
switch(conf-if-te-5/1/12)# dot1x reauthentication
switch(conf-if-te-5/1/12)# dot1x timeout re-authperiod 4000
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

6. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Configuring 802.1x port-control on specific interface ports

To configure 802.1x port-control on a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you want to modify.

1. Use the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. Use the **dot1x authentication** command to enable 802.1x authentication for the interface port.

```
switch(conf-if-te-5/1/12)# dot1x authentication
```

4. Change the port authentication mode to **auto**, **force-authorized** or **force-unauthorized**.

```
switch(conf-if-te-5/1/12)# dot1x port-control overlay auto
```

5. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

6. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Reauthenticating specific interface ports

To reauthenticate a supplicant connected to a specific interface port, perform the following steps from privileged EXEC mode. Repeat this task for each interface port you wish to reauthenticate.

1. Use the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. Use the **dot1x reauthenticate** command to re-authenticate a port where dot1x is already enabled.

```
switch(conf-if-te-5/1/12)# dot1x reauthenticate
```

4. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

5. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Disabling 802.1x on specific interface ports

To disable 802.1x authentication on a specific interface port, perform the following steps from privileged EXEC mode.

1. Enter the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. Use the **no dot1x port-control** command to disable 802.1x authentication.

```
switch(conf-if-te-5/1/12)# no dot1x port-control
```

4. Return to privileged EXEC mode.

```
switch(conf-if-te-5/1/12)# end
```

5. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Disabling 802.1x globally

To disable 802.1x authentication globally, perform the following steps from privileged EXEC mode.

1. Enter the **configure** command to change to global configuration mode.

```
switch# configure
```

2. Use the **no dot1x enable** command to disable 802.1x authentication.

```
switch(config)# no dot1x enable
```

3. Return to privileged EXEC mode.

```
switch(config)# end
```

4. Save the *running-config* file to the *startup-config* file.

```
switch# copy running-config startup-config
```

Checking 802.1x configurations

To check 802.1x configurations, perform the following steps from privileged EXEC mode.

1. To view all dot1x configuration information, use the **show dot1x** command with the **all** keyword.

```
switch# show dot1x all
```

2. To check 802.1x configurations for specific interface ports, use the **interface** command to select the interface port to modify.

```
switch(config)# interface tengigabitethernet 5/1/12
```

3. To check 802.1x authentication statistics on specific interface ports, use the **show dot1x** command with the **statistics interface** keyword.

```
switch# show dot1x statistics interface tengigabitethernet 5/1/12
```

4. To check all diagnostics information of the authenticator associated with a specific interface port, use the **show dot1x** command with the **diagnostics interface** keyword.

```
switch# show dot1x diagnostics interface tengigabitethernet 5/1/12
```

5. To check all statistical information of the established session, use the **show dot1x** command with the **session-info interface** keyword.

```
switch# show dot1x session-info interface tengigabitethernet 5/1/12
```

sFlow

sFlow protocol overview

The sFlow protocol is an industry-standard technology for monitoring high-speed switched networks. The sFlow standard consists of an sFlow agent that resides anywhere within the path of the packet and an sFlow collector that resides on a central server. This release is compliant with sFlow Version 5.

The sFlow agent combines the flow samples and interface counters into sFlow datagrams and forwards them to the sFlow Collector at regular intervals. The datagrams consist of information on, but not limited to, packet header, ingress interfaces, sampling parameters, and interface counters. Packet sampling is typically performed by the ASIC. The sFlow collector analyzes the sFlow datagrams received from different devices and produces a network-wide view of traffic flows. You can configure up to five collectors, using both IPv4 and IPv6 addresses.

The sFlow datagram provides information about the sFlow version, its originating agent's IP address, a sequence number, one sample, and protocol information.

The sFlow agent uses two forms of operation:

- Time-based sampling of interface counters
- Statistical sampling of switched packets

Be aware of the following limitations:

- If both port-based sampling and flow-based sampling are enabled on an interface, samples are based on port rate only.
- Port-based and flow-based sFlow are not supported on port channels, FCoE ports, or ISL ports.

Interface flow samples

A flow sample is based on random packets being forwarded to the sFlow collector at defined numeric intervals, either for the entire Brocade switch or for a single port interface. For example, every 4,096th packet is forwarded to the sFlow collector for analysis and storage.

The sampling rate is adaptive, and the sFlow agent is free to schedule the sampling to maximize internal efficiency.

NOTE
This type of random sampling provides estimated flow rates, but not perfect accuracy.

Packet counter samples

A polling interval defines how often the sFlow octet and packet counter for a specific interface are sent to the sFlow collector, but the sFlow agent is free to schedule the polling in order to maximize internal efficiency.

Hardware support matrix for sFlow

The following table identifies Brocade VDX 8770 and VDX 67xx/69xx device support for specific sFlow features.

TABLE 41 sFlow feature support

Feature	Brocade VDX 8770	Brocade VDX 67xx/69xx
sFlow global configurations for enabling sFlow, polling interval, collector, and sample rate	All are supported	All are supported
sFlow data source interface	Supports 1-Gbps, 10-Gbps, and 40-Gbps interfaces	Supports 10-Gbps interfaces only
sFlow data source: Front port trunks and VLANs	Not supported	Not supported
sFlow scanning for inbound, outbound, or both directions on a port	Supports inbound only	Supports inbound only
sFlow counter polling support on per-port, per-VLAN, or per-trunk basis	Supports only per-port counter polling	Supports only per-port counter polling
All standard **if_counters** and **Ethernet** counters	Supported	Supported
Multiple collector configuration	A maximum of five collectors can be configured.	A maximum of five collectors can be configured.
Extended Gateway, Extended router, and NAT/MPLS/URL header formats	Not supported	Not supported
Subagent-ID	Filled with slot number of the interface	Filled with a zero (0)
Agent IP address	Preference 1: Chassis IP Preference 2: Management CP IP	Management IP
Maximum packets per second	272 pkts/sec/ASIC VDX 8770-4: 6528 pkts/sec VDX 8770-8: 13056 pkts/sec	96 pkts/sec/ASIC
Sample rate calculation	Dropped packets (such as errors and ACL dropped packets) are not counted for the calculations used for sample generation	Dropped packets (such as errors and ACL dropped packets) are counted for the calculations used for sample generation
Maximum sFlow raw packet header size	228 bytes The hardware truncates the packet.	128 bytes The software truncates the packet.

Flow-based sFlow

Flow-based sFlow is used to analyze a specific type of traffic (flow based on access control lists, or ACLs). This involves configuring an sFlow policy map and binding it to an interface.

Flow-based sFlow considerations and limitations

The following considerations and limitations for flow-based sFlow should be kept in mind:

- A maximum of 16 profiles is allowed on Brocade VDX 8770 series platforms and 8 on a Brocade VDX 6740 series platforms.
- The purpose of flow-based sFLow is not to drop or trap packets on the basis of ACLs, but rather just to match traffic. Packets are still sampled and allowed, and sFlow samples are generated for any rule.
- Port-based sFlow takes precedence over flow-based sFlow. Both cannot operate simultaneously.
- On Brocade VDX 8770 series platforms, if a packet is classified as a Layer 3 IPv4 packet, with a match on destination address and IP type, and even if a packet's destination or source address matches a Layer 2 ACL, flow-based sFlow samples are not generated. This is not the case with Brocade VDX 6740 series platforms.
- Because all samples of different rates are sent to the collector from a single port, only the lowest sampling rate is used.

On Brocade VDX 6740 series platforms, if the traffic is Layer 2 then samples with a MAC ACL sample rate are collected. If the traffic is Layer 3, then samples with a Layer 3 ACL sample rate are collected.

Configuring the sFlow protocol

Consider the following topics when configuring the sFlow protocol.

Configuring the sFlow protocol globally

Brocade recommends that you globally configure sFlow on the Brocade switch first, and then enable sFlow on specific interface ports and make custom alterations, because sFlow parameters at the interface level can differ from those at the global level. For details, refer to Configuring sFlow for interfaces on page 264.

Enabling sFlow globally does not enable it on all interface ports. sFlow must be explicitly enabled on all the required interface ports. Refer to Enabling and customizing sFlow on specific interfaces on page 264.

For complete information on the sFlow CLI commands for the Brocade switch, refer to the *Network OS Command Reference*.

To configure sFlow globally, perform the following steps in global configuration mode.

1. Enter the **configure terminal** command to change to global configuration mode.

```
switch# configure terminal
```

2. Globally enable the sFlow protocol.

```
switch(config)# sflow enable
```

3. Designate the IP address (up to five addresses) for the sFlow collector server. Optionally, you can designate the port number.

NOTE
Both IPv4 and IPv6 addresses are supported. However, each address must be entered individually by means of a separate **sflow collector** command.

```
switch(config)# sflow collector 10.10.138.176 6343
switch(config)# sflow collector fd00::1900:4545:3:200:f8ff:fe21:67cf port 6343
switch(config)# sflow collector fd00::200:f8ff:fe21:67cf
```

4. Set the sFlow polling interval (in seconds).

```
switch(config)# sflow polling-interval 35
```

5. Set the sFlow sample-rate.

```
switch(config)# sflow sample-rate 4096
```

6. Return to privileged EXEC mode.

```
switch(config)# end
```

7. Confirm the sFlow configuration status by using the **show sflow** or **show sflow all** commands.

```
switch# show sflow
sFlow services are:                             enabled
Global default sampling rate:                   32768 pkts
Global default counter polling interval:        20 secs
Collector server address                        Number of samples sent
------------------------                        ------------------------------
fd00::1900:4545:3:200:f8ff:fe21:67cf            0
fd00::200:f8ff:fe21:67cf                        0
10.10.138.176                                   0

switch# show sflow all
sFlow services are:                             enabled
Global default sampling rate:                   32768 pkts
```

```
Global default counter polling interval:          20 secs
Collector server address                          Number of samples sent
------------------------                          ------------------------------
fd00::1900:4545:3:200:f8ff:fe21:67cf:6343         0
fd00:fd00::200:f8ff:fe21:67cf:                    0
10.10.138.176: 6343                               0
```

8. Clear any existing sFlow statistics to ensure accurate readings.

```
switch# clear sflow statistics
```

Configuring sFlow for interfaces

After the global sFlow configuration, sFlow must be explicitly enabled on all the required interface ports.

NOTE
When sFlow is enabled on an interface port, it inherits the sampling rate and polling interval from the global sFlow configuration.

Enabling and customizing sFlow on specific interfaces

Perform the following steps in privileged EXEC mode to enable and customize sFlow on an interface. This task assumes that sFlow has already been enabled at the global level; refer to Configuring the sFlow protocol globally on page 263.

1. Enter the **interface** command to specify the DCB interface type, the RBridge ID, and the slot/port number.

```
switch(config)# interface tengigabitethernet 22/0/16
```

2. Configure the sFlow polling interval.

```
switch(conf-if-te-22/0/16)# sflow polling interval 35
```

3. Use the **sflow enable** command to enable sFlow on the interface.

```
switch(conf-if-te-22/0/16)# sflow enable
```

4. Set the sFlow sample-rate.

```
switch(conf-if-te-22/0/16)# sflow sample-rate 8192
```

5. Confirm the sFlow configuration status on the specified interface.

```
switch# show sflow interface tengigabitethernet 22/0/16

sFlow info for interface TenGigabitEthernet 22/0/16
-----------------------------------------------
Configured sampling rate:         100 pkts
Actual sampling rate:             100 pkts
Counter polling interval:         100 secs
Samples received from hardware:   32
Port backoff-threshold :          272
Counter samples collected :       147
```

Configuring an sFlow policy map and binding it to an interface

Perform the following steps to configure an sFlow policy map and bind it to an interface.

1. Enter the **configure terminal** command to change to global configuration mode.

```
switch# configure terminal
```

2. Create a standard MAC access control list (ACL).

```
switch# mac access-list standard acl1
switch(conf-macl-std)# permit any
```

3. Create a class map and attach the ACL to the class map.

```
switch(conf-macl-std)# class-map class1
switch(config-classmap)# match access-group acl1
```

4. Create a policy map and attach the class map to the policy map.

```
switch(config-classmap)# policy-map policy1
switch(config-policymap)# class class1
```

5. Add an sFlow profile name by using the **map** command.

   This example assigns the profile name "policy1."

```
switch(config-policymap-class)# map sflow policy1
```

6. Bind the policy map to an interface.

```
switch(conf-if-te-1/8/1)# service-policy in policy1
```

Disabling sFlow on specific interfaces

NOTE
Disabling sFlow on the interface port does not completely shut down the network communication on the interface port.

To disable sFlow on a specific interface, perform the following steps in interface configuration mode.

1. Disable the sFlow interface.

```
switch(conf-if)# no sflow enable
```

2. Return to privileged EXEC mode.

```
switch(conf-if)# end
```

3. Confirm the sFlow configuration status on the specific interface.

```
switch# show sflow interface tengigabitethernet 5/0/12
```

Specifying the source of sFlow packets

You can specify the IPv4 or IPv6 address of either the chassis (virtual IP address) of the local Management Module as the source of sFlow packets, by means of the **sflow source-ip** command.

NOTE
The chassis virtual IP address is configured by means of the **chassis** command in RBridge ID configuration mode. The IP address of the local Management Module is configured by means of the **interface management** command in global configuration mode. By default, sFLow uses the chassis virtual IP address as the source of sFLow packets.

The following illustrates the use of the **sflow source-ip** command options.

1. Enter global configuration mode.

```
device# config
device(config)#
```

2. Use the **chassis-ip** keyword to specify the virtual IP address of the chassis as the source of the sFlow packets.

```
device(config)# sflow source-ip chassis-ip
```

3. Alternatively, use the **mm-ip** keyword to specify the IP address of the local Management Module as the source of the sFlow packets.

```
device(config)# sflow source-ip mm-ip
```

4. Do the following to verify the results of the above command.

```
device(config)# do show running-config sflfow
sflow enable
sflow source-ip mm-ip
```

5. Once a source type is specified, you can use the **no sflow source-ip** command to revert to the default configuration.

NOTE
Use the **do show running-config sflow** command in this configuration mode to confirm the configuration.

```
device(config)# no sflow source-ip
device(config)# do show running-config sflow
sflow enable
```

Enabling flow-based sFlow

Refer also to Flow-based sFlow on page 262.

Perform the following steps, beginning in global configuration mode.

NOTE
The "deny ACL" rule is not supported for flow-based sflow. Only the permit action is supported.

1. Create an sFlow profile. Be sure to specify the sampling-rate as a power of 2.

```
switch(config)# sflow-profile profile1 sampling-rate 256
```

2. Create a standard MAC ACL.

```
switch# mac access-list standard acl1
switch(conf-macl-std)# permit any
```

3. Create a class map and attach the ACL to the class map.

```
switch(conf-macl-std)# class-map class1
switch(config-classmap)# match access-group acl1
```

4. Create a policy map and attach the class map to the policy map.

```
switch(config-classmap)# policy-map policy1
switch(config-policymap)# class class1
```

5. Use the **map** command to add an sFlow profile name.

This example assigns the profile name "profile1."

```
switch(config-policymap-class)# map sflow profile1
```

6. Switch to interface configuration mode.

```
switch(config-policymap-class)# exit
switch(config)# interface ten 1/8/1
switch(conf-if-te-1/8/1)#
```

7. Bind the policy map to an interface.

```
switch(conf-if-te-1/8/1)# service-policy in policy1
```

Disabling flow-based sFlow on specific interfaces

To disable sFlow on a specific interface, perform the following steps in interface configuration mode.

NOTE
Disabling sFlow on an interface port does not completely shut down the network communication on the interface port.

1. Disable the sFlow interface.

```
switch(conf-if)# no sflow enable
```

2. Return to privileged EXEC mode.

```
switch(conf-if)# end
```

3. Switch to interface configuration mode.

```
switch(config-policymap-class)# exit
switch(config)# interface ten 1/8/1
switch(conf-if-te-1/8/1)#
```

4. Disable flow-based sFlow by removing the policy map.

```
switch(conf-if-te-1/8/1)# no service-policy in
```

5. Confirm the sFlow configuration status on the specific interface.

```
switch# show sflow interface tengigabitethernet 1/8/1
```

Configuring sFlow for VXLAN overlay gateway tunnels

Once an sFlow profile is created, you can apply it to a tunnel endpoint or all endpoints in VXLAN overlay-gateway configuration mode.

1. In global configuration mode, create an sFlow profile and set a sampling rate.

```
switch(config)# sflow-profile profile1 sampling-rate 256
```

NOTE
Sampling rates must be a power of 2. The default sampling rate is 32768 packets.

2. In global configuration mode, configure a VXLAN overlay gateway.

```
switch(config)# overlay-gateway gateway1
switch(config-overlay-gw-gateway1)#
```

3. In VXLAN overlay-gateway configuration mode, use the **sflow remote-endpoint** command with the profile name to specify the IPv4 address of a tunnel endpoint and add a VLAN.

```
switch(config-overlay-gw-gateway1)# sflow profile1 remote-endpoint 10.10.20.31 any vlan add 2000
```

4. Alternatively, you can specify all tunnel endpoints and a range of VLANs.

```
switch(config-overlay-gw-gateway1)# sflow profile1 remote-endpoint any vlan add 2000-3000
```

5. Use the **remove** keyword to remove VLANs.

```
switch(config-overlay-gw-gateway1)# sflow profile1 remote-endpoint any vlan remove 2000-2500
```

6. Use the **show sflow all** command to confirm the configuration

```
switch# show sflow all

sFlow services are:                     enabled
Global default sampling rate:           32768 pkts
Global default counter polling interval: 20 secs
Rbridge-Id              Collector server address          Number of samples sent
-------------------------------------------------------------------------------

sflow info for Tunnel
---------------------------------
ifindex: 1950351360
sFlow services are:           enabled
Samples received from hardware:       0
Effective sample-rate:                256
```

Switched Port Analyzer

Switched Port Analyzer protocol overview

The Switched Port Analyzer (SPAN) is used on a network switch to send a copy of network packets seen on one switch port to a network monitoring connection on another switch port. If you are interested in listening to or snooping on traffic that passes through a particular port, SPAN artificially copies the packets to a port connected to your analyzer. Usually, this traffic is limited to incoming or outgoing packets, but Network OS 4.0.0 and later allows bidirectional traffic monitoring on the source port.

SPAN in logical chassis cluster

SPAN in logical chassis cluster supports mirroring of a source port to a destination port lying on a different switch in the cluster. SPAN in logical chassis cluster is configured in the same manner, with the exception of the **source** command.

RSPAN

Remote SPAN, or RSPAN, extends SPAN by enabling remote monitoring of multiple switches across your network. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. The SPAN traffic from the sources is copied onto the RSPAN VLAN and then forwarded over trunk ports that are carrying the RSPAN VLAN to any RSPAN destination sessions monitoring the RSPAN VLAN.

NOTE
RSPAN is supported only on Brocade VDX 8770, VDX 6740, and VDX 6940 series platforms.

RSPAN consists of an RSPAN source interface and an RSPAN VLAN. The configured source port is mirrored to the RSPAN VLAN and the ports that are members of this VLAN receive the mirrored traffic.

All participating switches must be trunk-connected at Layer 2, and the remote VLAN must be configured on all the switches participating in the RSPAN session.

SPAN guidelines and limitations

Consider the following topics when configuring SPAN.

Standard SPAN guidelines and limitations

The following table lists the number of SPAN sessions with and without a shared destination port, as well as the number of SPAN sessions for both fabric-wide and VCS SPAN, with no sharing of the destination port.

NOTE
A SPAN session consists of either a single egress port, a single ingress port, or both.

TABLE 42 Number of SPAN sessions supported under various conditions

Brocade platform	Number of sessions without shared destination port	Number of sessions with shared destination port	Number of sessions, for both fabric-wide and VCS SPAN, without shared destination port
VDX 6740/6940 series	7	24	7 per box, 512 per VCS
VDX 8770 series	8 per port group, 48 per line card, 48 per chassis	No sharing needed (8 ports support 8 sessions)	48 per box, 512 per VCS

Brocade recommends that you be aware of the following additional standard guidelines for and limitations of SPAN connections:

- If there is congestion at the ingress span queue resulting from bandwidth-related back pressure at the ISL, the SPAN mirrored packets will be dropped and traffic will be lost.
- FCoE mirroring is not supported.
- For the traffic flowing across the switch, if the source port is in unknown mode (the node is in Layer 2 or Layer 3), then the untagged packets are dropped.
- The mirror port should not be configured to carry normal traffic.
- A port cannot be made a destination port for bidirectional mirroring if a different port supported by that ASIC is already configured as destination port for any type of mirroring.
- If a port is configured as a destination port for bidirectional mirroring, no other port supported by that ASIC can be made a destination port for any type of mirroring.
- The destination mirror port can handle from 1 to 40 Gbps (line rate) worth of mirror traffic, depending on the capability of the destination port. If multiple ports, or both flows on the same port, are mirrored to the same destination mirror port, then only the destination port's capacity worth of mirror traffic is mirrored and the remaining traffic is ignored.
- If the source port receives burst traffic and the destination mirror port cannot handle all the bursts, some of the burst traffic is not mirrored.
- Mirroring of Inter-Switch Link (ISL) ports is supported, but the destination port should reside on the same RBridge.
- Mirroring of LAG or port-channel interfaces is not supported, but LAG members can be mirrored.
- TRILL ports cannot be designated as a destination port.
- TRILL ports can be a source port, but mirroring is restricted to the port local to the source node ports.
- Ethernet Pause frames are not mirrored.
- Mirroring of trunk port is not supported, although the ASIC supports the mirroring of a trunk. To mirror a trunk, you must individually enable mirroring on all member ports.
- The multicast and broadcast statistics are correctly updated on TX ports for mirrored traffic.
- All commands except for **shutdown** and **no shutdown** are blocked on a destination mirror port.
- The interface counters are cleared when a port is successfully designated as a destination mirror port.
- The **show interface** command hides the Receive Statistics and Rate Info (Input) information for a destination mirror port.
- The MTU of a port should be set to the default value of 2500 bytes before it is made a destination mirror port. When the port is successfully designated as the destination mirror, the MTU of that port is automatically set to the maximum value of 9216 bytes. When the port becomes a non-destination mirror, the MTU is restored to the default value.
- Port mirroring is supported on any physical front-end user-configurable port. The source port can be part of a LAG, VLAG, VLAN, or any other user configuration
- A maximum of 512 mirror sessions are supported in logical chassis cluster and fabric cluster modes. A mirror session consists of either a single egress port, a single ingress port, or both.

RSPAN guidelines and limitations

The following configurations and restrictions for RSPAN should be kept in mind.

Basic considerations

The following table lists the number of RSPAN sessions with and without a shared destination port, as well as the number of SPAN sessions for both fabric-wide and VCS SPAN, with no sharing of the destination port.

TABLE 43 Number of RSPAN sessions supported under various conditions

Brocade platform	Number of sessions without shared destination port	Number of sessions with shared destination port	Number of sessions, for both fabric-wide and VCS SPAN, without shared destination port
VDX 6740/6740 series	7	24	7 per box, 512 per VCS
VDX 8770 series	8 per port group, 48 per line card, 48 per chassis	No sharing needed (8 ports support 8 sessions)	48 per box, 512 per VCS

Brocade recommends that you be aware of the following additional standard guidelines for and limitations of RSPAN connections:

- All participating switches must be connected by Layer 2 trunks.
- Inter-Switch Link (ISL) mirroring is not supported on RSPAN.
- The source and destination ports cannot both be TRILL (ISL) ports.
- RSPAN supports multi-hop.
- RSPAN can support both the fabric cluster and logical chassis cluster modes. However, using RSPAN in logical chassis cluster mode uses unnecessary ISL bandwidth because it floods the traffic on the ISL as well as the trunk port.
- If the source port is not Layer 2 and untagged traffic is mirrored, it will be dropped for RSPAN because untagged and unclassified traffic is dropped on an ISL trunk.
- On the Brocade VDX 6740 series platforms, if source port is in unknown mode, that is neither Layer 2 nor Layer 3, the packets are dropped and are not mirrored.
- Ethernet Pause frames are not mirrored.

VLAN considerations

- Before you configure an RSPAN session, you must create the RSPAN VLAN.
- A native VLAN cannot be made the RSPAN VLAN.
- The VLAN used for RSPAN should not be used for other purposes; furthermore, if the VLAN has ports as its members, it cannot be made an RSPAN VLAN. Only when the session is deconfigured, and the VLAN is deleted as an RSPAN VLAN, should the VLAN number be used for another purpose.
- You can configure any VLAN as an RSPAN VLAN as long as all participating network devices support the configuration of RSPAN VLANs and you use the same RSPAN VLAN for each RSPAN session in all participating network devices.
- You must configure the RSPAN VLANs on all source, intermediate, and destination network devices.
- Do not configure any ports in an RSPAN VLAN except the ports selected to carry RSPAN traffic. However, all configurations are allowed on the RSPAN destination port.
- The **vlan-id** of the packets marked for RSPAN will change to the RSPAN vlan-id.
- Access ports can be added to an RSPAN VLAN as destination ports.
- MAC address learning is disabled in the RSPAN VLAN.

Configuring SPAN

Refer also to Standard SPAN guidelines and limitations on page 269.

Configuring ingress SPAN

To configure SPAN for incoming packets only, do the following:

1. Open a monitor session and assign a session number.

```
switch(config)# monitor session 1
```

2. Configure the source port and the destination port, with the **rx** parameter for received packets.

   The destination port is always an external port.

```
switch(config-session-1)# source tengigabitethernet 1/0/15 destination tengigabitethernet 1/0/18
direction rx
```

   NOTE

   If the following error is displayed, use the **lldp disable** command in interface subtype configuration mode to disable LLDP on the destination port before preceding: `% Error: Destination port cannot be in L2/L3/Qos/ACL/802.1x/LAG member/Lldp/Port-profile/non-default-MTU`.

3. Optional: Use the **description** command to add a label to the monitor session.

```
switch(config-session-1)# description Hello World!
```

4. Optional: Repeat steps 1 and 2 as needed for additional ports.

   A monitor session can have only one source port. For additional ports you must create additional monitor sessions as needed for additional port mirroring sessions.

Configuring egress SPAN

To configure SPAN for outgoing packets only, do the following.

1. Open a monitor session and assign a session number

```
switch(config)# monitor session 1
```

2. Configure the source port and the destination port, with the **tx** parameter for transmitted packets.

   The destination port is always an external port.

```
switch(config-session-1)# source tengigabitethernet 1/0/15 destination tengigabitethernet 1/0/18
direction tx
```

   NOTE

   If the following error is displayed, use the interface **lldp disable** command to disable LLDP on the destination port before preceding: `% Error: Destination port cannot be in L2/L3/Qos/ACL/802.1x/LAG member/Lldp/Port-profile/non-default-MTU`.

3. Optional: Use the **description** command to add a label to the monitor session.

```
switch(config-session-1)# description Hello World!
```

4. Optional: Repeat steps 1 and 2 as needed for additional ports.

A monitor session can have only one source port. For additional ports you must create additional monitor sessions as needed for additional port mirroring sessions.

Configuring bidirectional SPAN

To configure SPAN for packets traveling in both directions, do the following.

1. Open a monitor session and assign a session number

```
switch(config)# monitor session 1
```

2. Configure the source port and the destination port, with the **both** parameter for all packets.

   The destination port is always an external port.

```
switch(config-session-1)# source tengigabitethernet 1/0/15 destination tengigabitethernet 1/0/18
direction both
```

   NOTE
   One of the following error messages may appear. If so, use the interface **lldp disable** command to disable LLDP on the destination port before preceding.

   - % Error: Destination port cannot have LLDP configuration on it.
   - % Error: Destination port cannot be in L2/L3/Qos/ACL/802.1x/LAG member/Lldp/Port-profile/non-default-MTU.

3. Optional: Use the **description** command to add a label to the monitor session.

```
switch(config-session-1)# description Hello World!
```

4. Optional: Repeat steps 1 and 2 as needed for additional ports.

   A monitor session can have only one source port. For additional ports you must create additional monitor sessions as needed for additional port mirroring sessions.

Deleting a SPAN connection from a session

To remove a single connection from a SPAN session, do the following.

1. Display the existing configuration of the monitor session.

```
switch# show monitor session 1
```

2. Open an existing monitor session.

```
switch(config)# monitor session 1
```

3. Use the **no** keyword to delete a particular port connection.

```
switch(config-session-1)# no source tengigabitethernet 1/0/15 destination tengigabitethernet 1/0/18
direction both
```

4. Display the monitor session again to confirm the deletion of the connection.

```
switch# show monitor session 1
```

Deleting a SPAN session

To remove a SPAN session, do the following:

1. Display the existing configuration of the monitor session.

```
switch# show monitor session 1
```

2. Delete the existing monitor session by using the **no monitor session** command.

```
switch(config)# no monitor session 1
```

3. Return to Privileged EXEC mode with the **exit** command.
4. Display the monitor session again to confirm the deletion of the connection.

```
switch# show monitor session 1
```

Configuring RSPAN

Refer also to RSPAN guidelines and limitations on page 271.

The principal difference between configuring SPAN and RSPAN is that RSPAN requires a remote VLAN to be created first, by means of the **rspan-vlan** command. This example demonstrates the configuration of a bidirectional RSPAN.

1. Create a remote VLAN on the destination interface.

```
switch(config)# interface vlan 100
```

2. Execute the **rspan-vlan** command to make the VLAN remote.

```
switch(config-vlan-100)# rspan-vlan
```

3. Exit the VLAN configuration mode.

```
switch(config-vlan-100)#end
```

4. Open a monitor session and assign a session number

```
switch(config)# monitor session 1
```

5. Configure the source port and the destination port, with the **both** parameter for bidirectional port mirroring.

   By modifying the direction parameter, you can control whether this is an ingress, egress, or a bidirectional SPAN.

   In the case of RSPAN, the destination is the VLAN, instead of a destination interface.

```
switch(config-session-1)# source tengigabitethernet 1/0/15 destination rspan-vlan 100 direction both
```

6. Optional: Use the **description** command to add a label to the monitor session.

```
switch(config-session-1)# description Hello World!
```

7. Use the **switchport** command to add a port to the RSPANVLAN to access the mirrored packets.

```
switch(config-session-1)# exit
switch (config)# interface ten 1/0/15
switch(conf-if-te-1/0/15)# switchport access rspan-vlan 100
```

8. Display the results of the configuration.

```
switch(conf-if-te-1/0/15)# do show vlan rspan-vlan
```

Flow-based SPAN and RSPAN

You can snoop on traffic that passes through a particular port, using flow-based SPAN or RSPAN to copy the packets to a port connected to the analyzer.

Flow-based SPAN selectively mirrors the traffic coming on the source port that matches an ACL-based filter to a destination port, which can be a local node or remote node to support RSPAN.

For example, assume there are two streams of traffic, one from the source Mac1 and other from source Mac2 are being forwarded from port te1/0/1 to port te1/0/2. You can, with the help of an ACL to permit only source Mac1 traffic, configure a flow-based SPAN session with the source on port te1/0/1 and port te1/0/2 as the destination port. All traffic coming in on port te1/0/1 originating from source Mac1 will be duplicated and sent to port te1/0/2. No mirroring occurs for traffic originating from source Mac2.

Consider the following guidelines and restrictions for flow-based SPAN and RSPAN:

- Flow-based SPAN source port cannot be an ISL port.
- Bi-directional association of the service policy cannot be supported with the current infrastructure. You must apply the service policy in both directions in two separate commands.
- Port-based or VXLAN-based SPAN sessions cannot be specified as the SPAN action.
- Deny rules in an service ACL is a pass through in flow-based QoS. Only permit rules with SPAN action result in flow-based SPAN.
- If a rule is configured as permit in flow-based ACL with SPAN action and the same rule is configured as deny in a user policy, the packet is dropped as per the user policy and the same is mirrored to the SPAN destination port.
- In a class map, if the SPAN action co-exists with any other QoS action (such as DSCP marking which results in frame editing), the mirrored packet is the original packet and hence does not reflect the frame editing done, as per the QoS action.

Configuring flow-based SPAN and RSPAN

You can replicate traffic from a defined source and direct it to snooping software on a designated port.

This task assumes you have already completed the following tasks:

- You have already created a policy map instance.
- You have already created a class map for the policy map.

To configure flow-based SPAN, perform the following task in privileged EXEC mode.

1. Enter global configuration mode.

```
switch# configure terminal
```

2. Create the monitor session.

```
switch(config)# monitor session 1
```

3. Set the destination port for the replicated traffic for SPAN.

NOTE
Remote ports are supported for RSPAN. Use the **rspan** *vlan_id* variable to configure for RSPAN.

```
switch(config)# destination rspan 100
```

4. Activate the pre-defined policy map.

```
switch(config)# policy-map policymap
```

5. Activate the pre-defined class for the policy map.

```
switch(config-policymap)# class policyclass
```

6. Activate the span session and assign it an identifying number.

```
switch(config-policyclass)# span session 1
```

7. Return to global configuration mode by executing the **exit** command twice.

```
switch(config-policyclass)# exit
switch(config-policymap)# exit
switch(config)#
```

8. Enter configuration mode for the source interface.

```
switch(config)#interface ten 1/0/1
```

9. Bind the policy to the interface.

```
switch(conf-te-1/0/1)# service-policy in policyclass
```

10. Confirm the session with the **show monitor** command.

```
switch# show monitor
Session                : 1
Type                   : Remote source session
Description            : [None]
Session Type: Flow based
Enabled on Source Interfaces:
**********************************
Ifname                                State      Direction
**********************************
gi1/0/2                                (Up)      Rx
te1/0/1                                (Up)      Rx
Destination Interface  : Vlan 100
```

Deleting the flow-based SPAN session

You remove the flow-based SPAN session by disassociating the span session from the policy-map . The pre-defined policy map and class as such are not deleted.

1. Activate the pre-defined policy map.

```
switch(config)# policy-map policymap
```

2. Unbind the session from the policy map.

```
switch(config)# no monitor session 1
```

3. Activate the pre-defined class for the policy map.

```
switch(config-policymap)# class policyclass
```

4. Deactivate the span session with the **no span session** version of the command and the identifying number.

```
switch(config-policyclass)# no span session 1
```

SFP Breakout Mode

SFP breakout overview

Breakout interfaces are those interfaces created on the breakout SFP. The number of interfaces created is dependent on the SFP type. For example, when a Quad SFP (QSFP) is not in breakout mode, only one 40-Gbps interface exists; however, when that QSFP has breakout mode enabled, four 10-Gbps interfaces are created. These interfaces, whether breakout mode is enabled or disabled, are administered and operate exactly the same as any other interface created on a regular SFP with no breakout capability. As a result, existing DCE module operations are not affected.

Fabric Inter-Switch Links (ISLs) are supported in breakout mode, and the default admin state for breakout interfaces is enabled.

In addition, beginning with Network OS 6.0.0, QSFP dynamic breakout is supported. This enables the user to configure breakout mode without having to reboot the switch.

Breakout mode support

Prior to Network OS 6.0.0, all platforms could support static breakout only. Beginning with Network OS 6.0.0, all ToR platforms are able to support dynamic breakout, which is more desirable. Only dynamic breakout is now supported on those platforms. All chassis platforms continue to support static breakout only. All transceiver types that supported static breakout continue to support dynamic breakout.

The table below lists the chassis platforms that support static breakout mode only.

TABLE 44 Chassis platforms supporting static breakout only

Platform	Port configuration	QSFP ports
VDX 8770-4 VDX 8770-8	12x40G and 27x40G	12 (12x40G) and 27 (27x40G)

The following table lists the ToR platforms that support dynamic breakout only, as well as the breakout-capable ports and the currently supported configuration.

TABLE 45 ToR platforms supporting dynamic breakout only

Supported platforms	Breakout-capable ports	Supported breakout configuration
VDX 2741	45-48 (4)	40G <-> 4x10G
VDX 2746	57-58 (2)	40G <-> 4x10G
VDX 6740	49-52 (4)	40G <-> 4x10G
VDX 6740T	49-52 (4)	40G <-> 4x10G
VDX 6740T-1G	49-52 (4)	40G <-> 4x10G
VDX 6940-36Q	1-36 (36)	40G <-> 4x10G
VDX 6940-144S	97-108 (12)	40G <-> 4x10G

Breakout mode properties

A breakout interface basically supports all operations or configurations that a regular interface supports (with few exceptions, which are noted in Breakout mode limitations on page 278). As such, it has the following properties:

- Has its own admin and operational state.
- Has its own ASIC resources interface statistics.
- Supports any configuration applicable to any regular SFP interface.
- Can be a port-channel or vLAG member.

Breakout mode can be static or dynamic, depending on the targeted platform. The default state for an SFP is "no breakout."

Ranging and the use of a comma delimiter are allowed in connector interface assignments.

The command **no connector** *rbridge-id/slot/port* is not allowed. Consequently, a connector assignment cannot be removed once it has been configured.

Breakout (or unbreakout) is not allowed under the following conditions:

- If any involved physical interfaces on a ToR switch is not in an administratively shut-down state.
- If an involved chassis line card is not in a powered-down state.
- If involved chassis ports on a 27x40G line card are not in Performance mode.
- If any involved physical interfaces belong to a port-channel.
- If any involved breakout port is configured as a FlexPort of Fibre Channel type.

Breakout mode interfaces

The SFP connector ID identifies a physical front-end SFP and has the same meaning as the port ID used in the interface name. The connector ID states which interfaces are created or deleted as a result of the breakout mode change. All interfaces attached to this connector must be disabled before the command is accepted. An interface created on an SFP in breakout-enabled mode adds a colon followed by a numeric suffix to the existing interface name.

This nomenclature identifies that a port is in breakout mode. For example, line card 2 port 1 on a node with RBridge ID 3 that has a Quad SFP (QSFP) in breakout disabled mode; if breakout is then set to enabled, the existing interface Fo 3/2/1 is deleted and four new interfaces —Te 3/2/1:1, Te 3/2/1:2, Te 3/2/1:3 and Te 3/2/1:4— are created. The new interfaces have the default port configuration. If breakout is set to disabled, the four Te interfaces are deleted and a single Fo interface is created, along with any configuration.

The table below shows an example of an SFP in breakout mode and its respective interface names.

TABLE 46 SFP breakout values

SFP # (rbridge-id/slot/port)	SFP type	Interface name	
		Breakout disabled	Breakout enabled
3/2/1	QSFP (4 x10G)	Fo 3/2/1	Te 3/2/1:1
			Te 3/2/1:2
			Te 3/2/1:3
			Te 3/2/1:4

Breakout mode limitations

In most circumstances, breakout interfaces behave the same as nonbreakout (normal) interfaces with regard to port attributes and states. Each breakout interface maintains its administrative state, operational state, and statistics. The exception is at the physical layer, whereby the hardware platform does not have per-breakout interface information.

- SFP media

  In breakout mode, there is only SFP and no per-breakout media information. The **show media** command displays the same media information for all breakout interfaces.

  NOTE
  The TX Power Field in the **show media** command is not supported by the 40G optics.

- LED state

  For Brocade VDX 6740 series platforms, the LED state for a breakout interface is deterministic. For all other supported platforms, the LED state for a breakout interface *is not* deterministic.

In addition, the 27x40G line card supports nine port groups of three ports each that you can configure for Performance or Density operating modes. If a port group is configured for Performance mode, the first two ports in a group are enabled in Performance mode, but the third port is disabled. If a port group is set for Density mode, all three ports operate in Density mode. Breakout mode can be configured only on ports enabled for Performance mode.

For more information on these modes, line card port groups, and instructions for configuring Performance mode on port groups, refer to the following sections in the *Brocade VDX 8770-4 Hardware Reference Manual* or *Brocade VDX 8770-8 Hardware Reference Manual*:

- 27x40 GbE operating modes
- Configuring operating modes on 27x40 GbE line cards

QSFP dynamic breakout

This feature allows the user to configure breakout mode without having to reboot the switch.

Prior to Network OS 6.0.0, the port breakout of a 40-Gbps QSFP port, typically used as an uplink port to aggregation switches, was available only in a static manner, requiring a reboot of the device and the line card to which the port belonged. Now such a port can be configured as four individual 10-Gbps ports without the need for a reboot, which also prevents other ports from being disturbed.

The following sections detail the effects of dynamic breakout and unbreakout, as well as special considerations for logical chassis cluster and fabric cluster modes.

Configuration updates with dynamic breakout

When dynamic breakout (or unbreakout) is configured, the following are deleted from the running configuration (and backend processes).

- For breakout, the original 40-Gbps physical interface (for example, FortyGigabitEthernet 1/0/50).
- For unbreakout, the four associated breakout 10-Gbps physical interfaces (for example, FortyGigabitEthernet 1/0/50:1-4).
- For unbreakout, the associated connector group and FlexPort configurations.
- All configurations under the physical interface context that is being removed.
- Other protocols (such as VLAN membership) that are associated with the physical interface context that is being removed.

When dynamic breakout (or unbreakout) is configured, the following are created in the running configuration.

- For breakout, a four-breakout 10-Gbps interface (for example, FortyGigabitEthernet 1/0/50:1-4) with a default configuration.
- For breakout, a default connector-group configuration for the platforms that support it.
- For unbreakout, the original 40-Gbps physical interface (for example, FortyGigabitEthernet 1/0/50) with a default configuration.

Considerations for logical chassis cluster mode

Note the following conditions for the preservation of a breakout configuration in this mode:

- The breakout configuration is preserved across a **copy default-config startup config** operation.
- The breakout configuration is not preserved across the operations of changes in RBridge ID, VCS ID, VCS mode, and a write erase.

Note the following conditions when a secondary node joins or rejoins a cluster:

- The breakout configuration in "enabled" state from the secondary node is always preserved.
- Following a rejoin with a nondefault local configuration, the breakout configuration in "enabled" state from the principal node is applied.
- For both ToR and chassis platforms, the breakout configuration from the principal switch is applied to the secondary switch. For dynamic-only (ToR) platforms, no reboot is needed. For static-only (chassis) platforms, the user is notified that a reboot is required for the configuration to take effect.

Considerations for fabric cluster mode

Note the following conditions for the behavior of a breakout configuration in this mode:

- The breakout configuration behaves like all other running configurations.
- The breakout configuration is preserved across a **copy running-config startup-config** operation.
- The breakout configuration is not preserved for the following conditions:
  - The execution of a **copy default-config startup-config** command
  - A reboot, unless the user issues the **copy running-config startup-config** command
  - Changes in RBridge ID, VCS ID, VCS mode, and a write erase

Configuring static breakout mode for a chassis system

To configure static breakout mode on a blade in a chassis, complete the following procedure.

1. Use the **linecard power-off** command to power off the appropriate line card.

```
device# power-off linecard 2
```

NOTE
The interface and its current configuration will still exist in the configuration, as revealed by a **show running-config** command, but operational commands will not show interfaces on this line card.

2. Enter hardware configuration mode.

```
device# configure terminal
device(config)# hardware
```

3. Enter connector configuration mode for the port you wish to break out.

```
device(config-hardware)# connector 2/2/1
```

4. Execute the **sfp breakout** command. For example, to break out a 40G port into four 10G ports you would use the following example.

```
device(config-connector-2/2/1)# sfp breakout
```

NOTE
Existing interfaces under the previous mode are removed, along with their associated configurations. If the target port is a QSFP, one FortyGigabitEthernet interface is deleted. The configurations of nontarget ports are not affected.

5. (Optional) You can also apply the command to a range of ports, or apply a comma delimiter, as in the following example.

```
device(config-hardware)# connector 2/2/1-3,5
device(config-connector-2/2/1-3,5)# sfp breakout
```

NOTE
The interfaces under the new mode are not created yet; they are created only when the line card is powered back on.

6. In fabric cluster mode, copy the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

7. Use the **power-on linecard** command to power the line card back on.

```
device# power-on linecard 2
```

NOTE
The SFP interfaces come up under the new mode, with default configurations. Unaffected interfaces retain the configurations they had before the line card was powered off.

8. Execute the **do show running-config hardware connector** command to confirm the breakout.

```
device(config-connector-2/2/1)# do show running-config hardware connector
hardware

 connector 2/2/1
   sfp breakout
 !
```

9. To unbreakout a connector and revert to the previous configuration, use the **no sfp breakout** command as in the following example. You must power off and power on the line card as in the previous steps.

```
device(config-connector-2/2/1)# no sfp breakout
```

NOTE
Four TenGigabitEthernet interfaces are deleted.

Configuring dynamic breakout mode for a ToR system

To configure dynamic breakout mode on a ToR system, complete the following procedure.

1. Ensure that the target physical port is administratively down.

```
device# configure terminal
device(config)# interface fortygigabitethernet 2/2/49
device(config-if-fo-2/2/49)# shut
```

2. Enter hardware configuration mode.

```
device# configure terminal
device(config)# hardware
```

3. Enter connector configuration mode for the port you wish to break out.

```
device(config-hardware)# connector 2/2/49
```

4. Execute the **sfp breakout** command. For example, to break out a 40G port into four 10G ports you would use the following example.

```
device(config-connector-2/2/49)# sfp breakout
```

NOTE

The interface comes up in the "no shut" state. You do not need to reboot the system.

5. (Optional) You can also apply the command to a range of ports, or apply a comma delimiter, as in the following example.

```
device(config-hardware)# connector 2/2/49-50,52
device(config-connector-2/2/49-50,52)# sfp breakout
```

6. Execute the **do show running-config hardware connector** command to confirm the breakout.

```
device(config-connector-2/2/49)# do show running-config hardware connector
hardware

 connector 2/2/49
   sfp breakout
 !
```

7. You can also confirm the entire configuration by using the **show ip interface brief** command.

```
switch# show ip int br
Interface                      IP-Address     Vrf                Status         Protocol
==========================     ==========     ================   =========      =========
FortyGigabitEthernet 2/2/50    unassigned     default-vrf        up down
FortyGigabitEthernet 2/2/51    unassigned     default-vrf        up down
FortyGigabitEthernet 2/2/52    unassigned     default-vrf        up down
TenGigabitEthernet 2/2/1       unassigned     default-vrf        up up (ISL)
TenGigabitEthernet 2/2/2       unassigned     default-vrf        up down
TenGigabitEthernet 2/2/3       unassigned     default-vrf        up down
TenGigabitEthernet 2/2/47      unassigned     default-vrf        up down
TenGigabitEthernet 2/2/48      unassigned     default-vrf        up down
TenGigabitEthernet 2/2/49:1    unassigned     default-vrf        up down(ISL)
TenGigabitEthernet 2/2/49:2    unassigned     default-vrf        up down(ISL)
TenGigabitEthernet 2/2/49:3    unassigned     default-vrf        up down(ISL)
TenGigabitEthernet 2/2/49:4    unassigned     default-vrf        up down(ISL)
Vlan 1                         unassigned     administratively   down down
Vlan 4093                      unassigned                        up down
Vlan 4095                      unassigned     administratively   down down
```

8. To unbreakout a connector and revert to the previous configuration, first ensure that the breakout ports are administratively down, and then use the **no sfp breakout** command on the connector, as in the following example.

   a) Shut down all of the breakout ports administratively.

```
device# configure terminal
device(config)# interface tengigabitethernet 2/2/49:1-4
device(config-if-te-2/2/49:1-4)# shut
```

   b) Disable breakout for the connector.

```
device(config)# hardware
device(config-hardware)# connector 2/2/49
device(config-connector-2/2/49)# no sfp breakout
```

NOTE

Four TenGigabitEthernet interfaces are deleted. The 40G interface is created automatically with a default configuration in the "no shut" state.

Reserving and releasing breakout ports

You can use the **dpod** command to enable the Dynamic Ports on Demand feature for breakout ports, and reserve or release them while they are in breakout mode.

1. In global configuration mode, execute the **dpod** command with the **reserve** keyword for the interface you want to reserve.

```
device(config)# dpod 2/2/49 reserve
```

NOTE
You cannot reserve a subinterface.

2. To release the interface, use the **release** keyword.

```
device(config)# dpod 2/2/49 release
```

Configuring Dual Personality Ports

Dual Personality Ports overview

The dual personality port feature for the Brocade VDX 6940-144S allows ports 97, 98, 103, and 104 on this device to be configured as 40 GbE QSFP+ or 100 GbE QSFP28 ports provided appropriate transceivers are installed.

Four rows of 40 GbE ports are located on the right side of the Brocade VDX 6940-144S front panel with three ports per row. Each row is assigned a dual personality group number of 1 through 4. The following table shows port group mapping to the dual personality port and other ports in the row. Refer to the *Brocade VDX 6940 Hardware Installation Guide* for the specific location of dual personality ports and port groups on the device.

TABLE 47 Dual personality port groups

Port Group	40/100 GbE port # (dual personality port)	40 GbE port #	40 GbE port #
1	97	99	101
2	98	100	102
3	103	105	107
4	104	106	108

The group number is used in Network OS commands for configuration. When you configure 100 GbE mode for a port group, only the dual personality port operates at 100 GbE, if the appropriate 100 GbE QSFP28 transceiver is installed in that port. The remaining two 40 GbE ports in the group are disabled. Conversely, if you reconfigure the port group from 100 GbE to 40 GbE operation, the dual personality port will transition to 40 GbE mode. All three ports in the group will come back online at 40 GbE operation if appropriate 40 GbE optics transceivers installed.

The following port configurations are possible for the twelve 40 GbE ports on the device:

- 12 40 GbE QSFP ports and No 100 GbE QSFP28 ports
- 9 40 GbE QSFP ports and 1 100 GbE QSFP28 port
- 6 40 GbE QSFP ports and 2 100 GbE QSFP28 ports
- 3 40 GbE QSFP ports and 3 100 GbE QSFP28 ports
- No 40 GbE QSFP ports and 4 100 GbE QSFP28 ports

NOTE
100 GbE QSFP28 transceivers do not support breakout mode.

Configure 100 GbE operation on specific dual personality port groups using the **port-group** *rbridge-id/slot/port-group-id* and **mode** *100g* commands. You can transition to 40 GbE operation for specific port groups using the **port-group** *rbridge-id/slot/port-group-id* and **mode** *40g* commands.

NOTE
After configuring 100 GbE or 40 GbE operation for a port group, you must reboot the Brocade VDX 6940-144S to enable the new operation mode.

The **show hardware port-group** command displays the port numbers mapped to port groups and modes currently configured for each group. Following is an example showing port group 3 enabled for 100 GbE mode and groups 1, 2, and 4 enabled for 40 GbE mode.

```
device# show hardware port-group

Port-Group          Ports             Mode
==============================================
237/0/1       237/0/97 ,99 ,101        40G
237/0/2       237/0/98 ,100,102        40G
237/0/3       237/0/103,105,107        100G
237/0/4       237/0/104,106,108        40G
```

Before enabling 100 GbE mode for a port group, perform the following tasks:

- Disable 4x10 GbE SFP breakout mode for all ports in that port group, if configured. Ports must be enabled for 40 GbE mode.
- Disable all interfaces in the port group.
- Install the appropriate 100 GbE QSFP28 transceiver in the dual-personality port cage for that port group (port 97, 98, 103, or 104).
- Reserve the DPOD for the ports in the port group.

Limitations and considerations

Consider the following when configuring and using dual-personality ports and port groups:

- When you enable 100 GbE mode for a port group, the leftmost port in the group (dual personality port 97, 98, 103, or 104) will be enabled for 100 GbE operation and the other ports in the port group will be disabled.
- When a port group is enabled in 40 GbE mode, all three ports can be configured to breakout mode. However, you must disable breakout mode on all three ports before configuring 100 GbE mode for the port group. When a port group is enabled in 100 GbE mode, the dual personality port will enabled for 100 GbE operation and the other two ports will be disabled.
- Ports with supported 40 GbE QSFP transceivers that are not in a dual personality port group configured for 100 GbE operation can function in 40 GbE mode and can be configured 4x10 GbE breakout mode.
- To transition dual personality ports 97, 98, 103, or 104 from 100 GbE to 40 GbE operation, you must install a supported 40 GbE QSFP+ transceiver in that port. Note that you can still configure 40 GbE mode for the port group, but without a 40 GbE QSFP+ transceiver installed, the dual personality port will be disabled. The other two ports in the group will operate at 40 GbE. This is same case when configure from 40 GbE mode to 100 GbE mode.
- After configuring 100 GbE or 40 GbE operation for a port group, you must reboot the system to enable the new operation mode.
- Any configuration applied to ports in a port group will be removed when you change to 100 GbE mode or 40 GbE mode.
- A specific dual personality port license is not required, but to enable a dual-personality port in a port group for 100 GbE operation, all three ports in the port group must have ports on demand (POD) reservations allocated from a 40 GbE Port Upgrade license.
- Trunking is not supported on 100 GbE ports.

Configuring 100 GbE operation

Perform the following tasks before enabling 100 GbE operation:

- Disable 4x10 GbE breakout mode for all 40 GbE ports in that port group.
- Disable all interfaces in the port group using the **shutdown** command.
- Install a qualified 100 GbE transceiver in the dual personality port (the leftmost port in the port group row).

To configure a dual personality port for 100 GbE operation, use the following steps. In the following example, port 97 in dual personality port group 1 is configured for 100 GbE operation.

1. Enter hardware configuration mode.

```
device# configure terminal
Entering configuration mode terminal
device(config)# hardware
```

2. Disable all ports in port group 1 using the **shutdown** command.
3. Enter the port group configuration mode for dual personality port group 1.

```
device(config-hardware)# port-group 1/0/1
device(config-port-group-1/0/1)#
```

NOTE
The **port-group** command uses variable *rbridge-id/slot/port-group-id* to identify the port. To configure a dual personality port group, use slot=0 and port-group-id=1-4. Slot 0 is always used for fixed switches, such as the Brocade VDX 6940-144S.

4. Configure the port group for 100 GbE operation using the **mode 100g** command.

```
device(config-port-group-1/0/1)# mode 100g
```

5. In fabric cluster mode, copy the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

6. Reboot the Brocade VDX 6940-144S to enable the new operating mode.
7. Execute the **show running-config hardware port-group** command to confirm the operating mode for the port group.

```
device# show running-config hardware port-group
hardware
port-group 1/0/1
  mode 100g
```

8. Execute the **show hardware port-group** command to display the port numbers mapped to port groups and modes defined for each group.

```
device# show hardware port-group

Port-Group          Ports              Mode
==============================================
237/0/1      237/0/97,99,101          100G
237/0/2      237/0/98,100,102         40G
237/0/3      237/0/103,105,107        40G
237/0/4      237/0/104,106,108        40G
```

Configuring 40 GbE operation

Perform the following tasks before enabling 40 GbE operation:

- Disable all interfaces in the port group using the **shutdown** command.
- Install a qualified 40 GbE transceiver in the dual personality port (the leftmost port in the port group row).

To transition a dual-personality port from 100 GbE to 40 GbE operation, use the following steps. Steps in the following example configure port 97 in dual personality port group 1 for 40 GbE operation.

1. Enter hardware configuration mode.

```
device# configure terminal
Entering configuration mode terminal
device(config)# hardware
```

2. Disable all ports in port group 1 using the **shutdown** command.
3. Enter the port group configuration mode for dual personality port group 1.

```
device(config-hardware)# port-group 1/0/1
device(config-port-group-1/0/1)#
```

> NOTE
> The **port-group** command uses variable *rbridge-id/slot/port-group-id* to identify the port. To configure a dual personality port group, use slot=0 and port-group-id=1-4. Slot 0 is always used for fixed switches, such as the Brocade VDX 6940-144S.

4. Configure the port group for 40 GbE operation using the **mode 40g** command.

```
device(config-port-group-1/0/1)# mode 40g
```

5. In fabric cluster mode, copy the running configuration to the startup configuration.

```
device# copy running-config startup-config
```

6. Reboot the Brocade VDX 6940-144S to enable the new operating mode.
7. Execute the **show running-config hardware port-group** command to confirm the operating mode for the port group.

```
device# show running-config hardware port-group
hardware
port-group 1/0/1
  mode 40g
```

8. Execute the **show hardware port-group** command to display the port numbers mapped to port groups and modes defined for each group.

```
device# show hardware port-group

Group No.      Port Number          Current Mode
=========================================
0/1            1/0/97,99,101         40G
0/2            1/0/98,100,102        40G
0/3            1/0/103,105,107       40G
0/4            1/0/104,106,108       40G
```

FlexPort

FlexPort overview

The FlexPort feature allows up to 32 ports to transmit data as either 10G Ethernet or Fibre Channel, and to be changed from one type to the other without requiring a reboot. These ports are grouped together as connector groups.

Connector groups share common speed and protocol type properties. The settings allow any port within each connector group to operate as either Ethernet or Fibre Channel ports, and support the appropriate optic transceivers. The Fibre Channel ports must be running any supported 8-Gbps or 16-Gbps Brocade FC transceivers.

The default setting is for Ethernet, and ports that do not support the Fibre Channel protocol are not allowed to have their connector group setting changed from the default setting.

Speed combinations allowed per connector-group are:

- **LowMixed** - 2/4/8G Fibre Channel, and Ethernet speeds (default)
- **HighMixed** - 16G Fibre Channel, and Ethernet speeds
- **FibreChannel** - 2/4/8/16G Fibre Channel (Only if all eight ports are already set as Fibre Channel type)

For the currently supported platforms, the connector-group numbers range from 1 through 6. They are related directly to the ports as numbered on each platform. The connector-group numbers that are allowed to be changed and their associated port numbers are shown in the table below. For example, on a Brocade VDX 2741, ports 43 through 50 belong to connector group 1. Not every connector group is supported on a switch.

> **ATTENTION**
> Changing connector-group speed is disruptive to ports within the same connector-group.

TABLE 48 FlexPort supported hardware

Platform	Port number range	Connector group
Brocade VDX 2741	57-58	3
Brocade VDX 6740	1-8	1
	17-24	3
	33-40	5
	41-48	6

> **NOTE**
> For the Brocade VDX 6740T and Brocade VDX 6740T-1G, Ethernet operation is supported on 40-GbE QSFP ports configured in 40-GbE mode that use qualified 40-GbE transceivers and on 40-GbE ports in 4x10 GbE breakout mode that use qualified 4x10-GbE QSFP transceivers. Fibre Channel operation is supported on 40-GbE ports configured in 40-GbE breakout mode that use qualified 4x16 QSFP+ transceivers. FlexPort is not supported on 40-GbE QSFP ports on the Brocade VDX 6740.

Only Brocade-branded SFPs are supported. While all 2/4/8/16G Brocade-branded SFPs are allowed for Brocade engineering purposes, only the five SFPs listed below are qualified and supported:

- 8G SWL
- 8G LW-10KM
- 8G ELWL-25KM
- 16G-SWL
- 16G-LW-10KM

Configuring FlexPort

The FlexPort feature allows up to 32 ports to transmit either Ethernet or Fibre Channel.

The FlexPort feature is set to Ethernet by default. You should only need to perform this task to switch to Fibre Channel, or back to Ethernet.

1. Enter hardware configuration mode.

```
switch(config)#hardware
switch(config-hw)#
```

2. Enter FlexPort configuration mode for the switch. This command configures FlexPort 5 on RBridge ID 1. FlexPort 5 is part of connector group 1 on the Brocade VDX 6740.

```
switch(config-hw)# flexport 1/0/5
switch(conf-hw-flex-1/0/5)#
```

3. Set the FlexPort type to Fibre Channel.

```
switch(conf-hw-flex-1/0/5)# type FibreChannel
```

4. Optionally, you may adjust the speed for the connector group. For example, if you want the connector group 1 to function at 16Gbps speed:

```
switch(conf-hw-flex-1/0/5)# connector-group 1/0/1
switch(config-connector-group-1/0/1)# speed HighMixed
```

5. Repeat these steps for additional switches as needed.
6. Confirm the FlexPort configuration with the **show running-config hardware flexport** command.

```
switch# show running-config hardware flexport
Hardware
 flexport 1/0/1
    type FibreChannel
!
Hardware
flexport 1/0/5
    type FibreChannel
!
connector-group 1/0/1
  speed HighMixed
!
```

FlexPorts and breakout mode

Breakout mode is supported on FlexPorts. This allows you to breakout certain QSFP and SFP ports and switch them between Fibre Channel and Ethernet without the need to reboot the device.

NOTE
A 40G license is not required to enable breakout Fibre Channel FlexPorts.

FlexPort requires the QSFP port to be configured in breakout mode.

Restrictions on FlexPorts in breakout mode:

- Fibre Channel SFP+ ports can operate at 2G, 4G, 8G and 16G speeds.
- Fibre Channel QSFP+ ports can operate at 4G, 8G and 16G.

TABLE 49 Valid breakout ports for FlexPort

Platform	Port number range	Connector group
Brocade VDX 2741	43-50 51-56 57-58 (Ports 43 through 56 are SFP and 57 through 58 are QSFP.)	1 2 3
Brocade VDX 6740T Brocade VDX 6740T-1G	49-50 51-52	7 8

Configuring FlexPorts for breakout mode

Configures FlexPorts to function in breakout mode.

The FlexPort feature is set to Ethernet by default.

1. Enter hardware configuration mode.

```
switch(config)#hardware
switch(config-hw)#
```

2. Enter FlexPort configuration mode for the switch. This command configures FlexPort 49:1 on RBridge ID 1. Flexport 49:1 is part of connector group 7 on the Brocade VDX 6740.

```
switch(config-hw)# flexport 1/0/49:1
switch(conf-hw-flex-1/0/49:1)#
```

3. Set the FlexPort type to Fibre Channel.

```
switch(conf-hw-flex-1/0/49:1)# type FibreChannel
```

4. Optionally, you may adjust the speed for the connector group. For example, if you want the connector group 7 to function at 16 Gbps:

```
switch(conf-hw-flex-1/0/49:1)# connector-group 1/0/7
switch(config-connector-group-1/0/49:1)# speed HighMixed
```

5. Repeat these steps for additional switches as needed.

6. Confirm the FlexPort configuration with the **show running-config hardware flexport** command.

```
switch# show running-config hardware flexport
Hardware
 flexport 1/0/1
    type FibreChannel
!
Hardware
flexport 1/0/49:1
    type FibreChannel
!
connector-group 1/0/7
  speed HighMixed
!
```

Link-State Tracking (LST)

LST overview

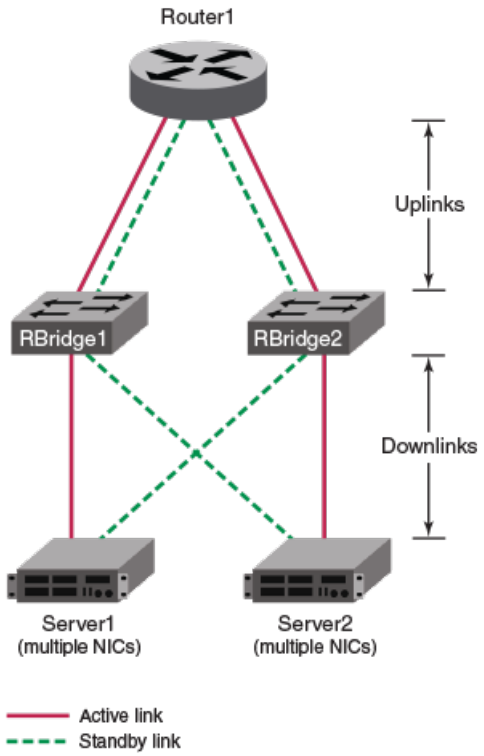
Link-state tracking (LST) is a relationship that you can configure and enable for redundant-link networking topology, to prevent traffic loss between upstream and downstream RBridge links.

Redundant-link topology

In a redundant-link topology, an alternate, standby network path is defined. If the primary, active path becomes unavailable, network traffic is rerouted to the standby path.

The following is a basic redundant-link topology:

FIGURE 39 A redundant-link topology



The primary path from the NICs of Server1 to Router1 is through RBridge1, and the alternate path is through RBridge2.

The primary path from the NICs of Server2 to Router1 is through RBridge2, and the alternate path is through RBridge1.

Redundant-link topology can include the following options:

- Links can be any of the following types:
  - Single physical port
  - Multiple physical ports
  - Port-channels
- VCS fabrics can be in place of or in addition to the individual RBridges. For more details, refer to VCS redundant-link topology on page 299.

If an upstream device stops functioning, there is a danger that the RBridge will continue to forward traffic upstream on the primary path, with ensuing loss of data.

LST operation

When link-state tracking (LST) is enabled, if an upstream link goes down, the downstream link automatically shuts down. At that point, server traffic is routed through the standby path, with negligible traffic loss. If the primary upstream link returns, LST restores the primary downstream link and reroutes the upstream traffic through the primary path.

The following upstream issues are among those that LST can handle:

- Failure of an upstream device
- Cable disconnection
- Other link failure

Default response to uplink failure

As indicated in the following table, LST supports various combinations of uplinks and downlinks. By default, if one or more tracked uplinks fail, LST shuts all downlinks.

TABLE 50 Default response to uplink failure

Downstream links	Upstream links	Default response to uplink failure
1	1	If the uplink fails, LST shuts the downlink.
n	1	If the uplink fails, LST shuts all downlinks.
1	n	If one or more uplinks fail, LST shuts the downlink.
n	n	If one or more uplinks fail, LST shuts all downlinks.

NOTE

When considering the number of uplinks or downlinks, a port-channel is equivalent to a physical port.

Response to uplink failure with min-link specified

For multiple uplinks, the **track** command **min-link** option enables you to modify the default LST response to uplink failure.

By default, if one or more tracked uplinks fail, LST shuts all downlinks that track those uplinks. But if you specify **min-link**, such downlinks remain open if the number of functioning uplinks is equal to or greater than **min-link**.

As indicated in the following table, for multiple uplinks the default response is modified by the **min-link** value.

TABLE 51 Response to uplink failure with min-link specified

Downstream links	Upstream links	Response to uplink failure with min-link specified
1	1	If the uplink fails, LST shuts the downlink.
n	1	If the uplink fails, LST shuts all downlinks.
1	n	If the number of functioning uplinks < min-link, LST shuts the downlink.
n	n	If the number of functioning uplinks < min-link, LST shuts all downlinks.

General configuration guidelines for LST

The following are general configuration guidelines for link-state tracking (LST):

- LST is supported only for the following Layer 2 or Layer 3 interface types:
  - Physical interfaces (including breakout ports)
  - Port channels
    - However, port channels are not supported for LST under FlexPort.
    - For a port-channel downlink, if an upstream link goes down, LST shuts down the member physical ports of the port channel, but not the port channel.
- The following are examples of interface types for which LST is NOT supported:
  - 100 Mbps ports
  - Fibre-channel (FC)
  - Management (including loopback)
  - Switch virtual interface (SVI)
- For multiple uplinks and multiple downlinks, LST is supported also for configurations that include both Layer 2 and Layer 3 ports.
- You cannot configure a given port both as an LST uplink and downlink.
- In general, LST is configured on primary links, but not on standby links.
- You can implement LST on multiple hops in a network.
- You can implement LST on forward-referenced port channels (LAG ports with no member ports).
- LST operates on the operational level of uplinks rather than their STP forwarding state. So the following are required:
  - The redundant network must be loop-free.
  - If STP or RSTP is enabled, every uplink under LST must be in STP forwarding state.

  NOTE
  If needed, refer also to LST configuration guidelines under VCS on page 300.

Configuring LST for independent RBridges

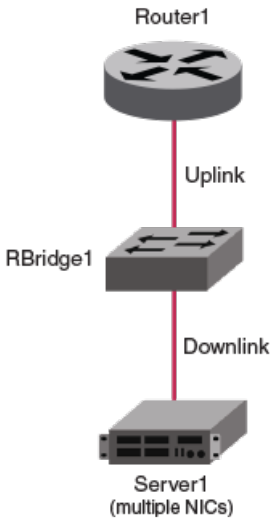
These topics explain how to configure link-state tracking (LST) for non-VCS topologies.

Configuring LST for single-link topologies

Use this procedure to configure LST on an RBridge with one uplink and one downlink.

In the following diagram of a single-link network topology, any of the links can be either physical port or port-channel.

FIGURE 40 Single-link topology



1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 1/0/1
```

3. Enter the **track interface** command to configure tracking for an uplink interface.

```
device(conf-if-te-1/0/1)# track interface ethernet 1/0/20
```

4. Enter the **track enable** command to enable tracking.

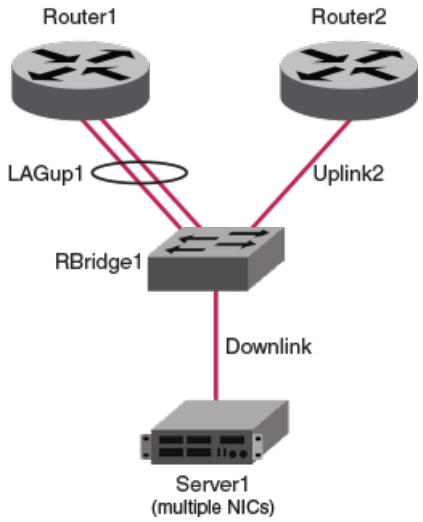
```
device(conf-if-te-1/0/1)# track enable
```

Configuring LST for multiple-uplink toplogies

Use this procedure to configure LST on an RBridge with one downlink and two or more uplinks.

In the following diagram of a single-downlink, multiple-uplink topology, any of the links can be either physical port or port-channel.

FIGURE 41 Single-downlink, multiple uplink topology



1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 3/0/8
```

3. For each uplink interface that you want to track, enter the **track interface** command.

```
device(conf-if-te-3/0/8)# track interface ethernet 3/0/18
device(conf-if-te-3/0/8)# track interface ethernet 3/0/19
```

4. To modify the default behavior (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(conf-if-te-3/0/8)# track min-link 1
```

   In this case, the downlink shuts down only if all of the uplinks are down. If only one of the two uplinks are down, the downlink stays open.

5. Enter the **track enable** command to enable tracking.

```
device(conf-if-te-3/0/8)# track enable
```

Configuring LST for multiple downlink/uplink topologies

Use this procedure to configure LST on an RBridge with multiple downlinks and multiple uplinks.

Perform this procedure on every downlink interface for which you are implementing LST. For example, on Switch3 in the below diagram, perform this procedure on the following interfaces:

- LAGdown1
- Downlink2

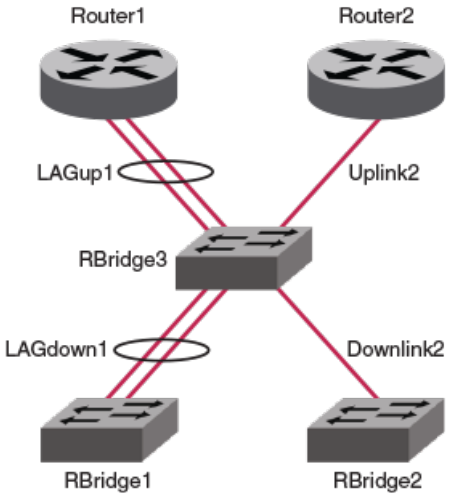
The code examples in this procedure are performed on LAGdown1, which tracks two uplink interfaces:

- LAGup1 (a port-channel)

- Uplink2 (a physical port)

In the following diagram of a multiple-downlink, multiple-uplink topology, any of the links can be either physical port or port-channel.

FIGURE 42 Multiple downlink/uplink topology



1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 3/0/8
```

3. For each uplink interface that you want to track, enter the **track interface** command.

```
device(conf-if-te-3/0/8)# track interface ethernet 3/0/18
device(conf-if-te-3/0/8)# track interface port-channel 1
```

4. To modify the default behavior (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(conf-if-te-3/0/8)# track min-link 1
```

   In this case, the downlink shuts down only if all of the uplinks are down. If only one of the two uplinks are down, the downlink stays open.

5. Enter the **track enable** command to enable tracking.

```
device(conf-if-te-3/0/8)# track enable
```

Configuring LST for VCS fabrics

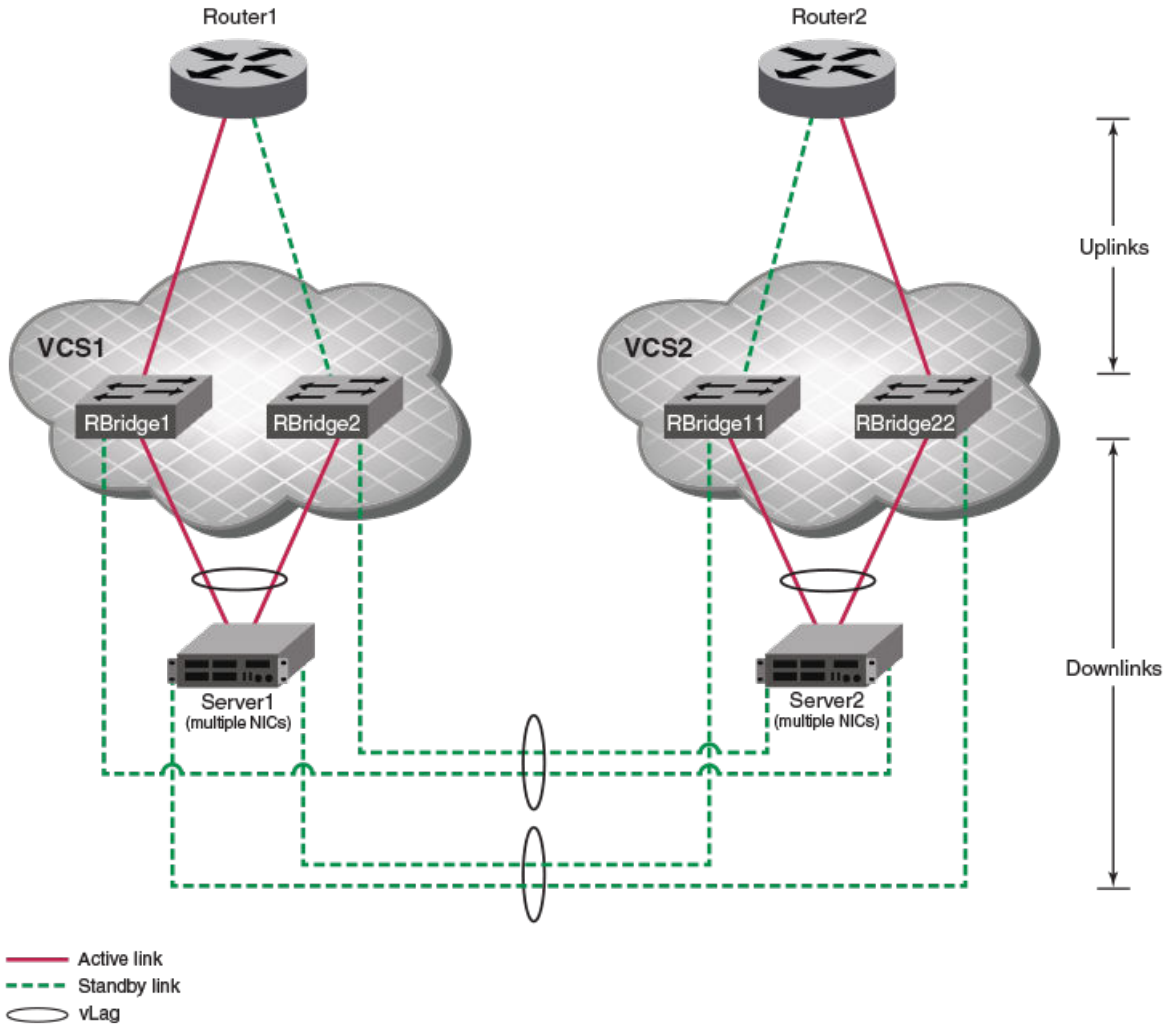
These topics explain how to configure link-state tracking (LST) for topologies that include one or more VCS fabrics.

VCS redundant-link topology

In a redundant-link topology, an alternate, standby network path is defined. If the primary, active path becomes unavailable, network traffic is rerouted to the standby path. This section deals with redundant-link topologies that include one or more VCS fabrics.

The following is a redundant-link topology that includes multiple VCS fabrics.

FIGURE 43 Redundant-link topology for VCS fabrics



The primary path from the NICs of Server1 to Router1 is through VCS1, and the alternate path is through VCS2.

The primary path from the NICs of Server2 to Router1 is through VCS2, and the alternate path is through VCS1.

LST configuration guidelines under VCS

The following are additional configuration guidelines for LST on RBridges in VCS fabrics.

NOTE
Refer also to General configuration guidelines for LST on page 295.

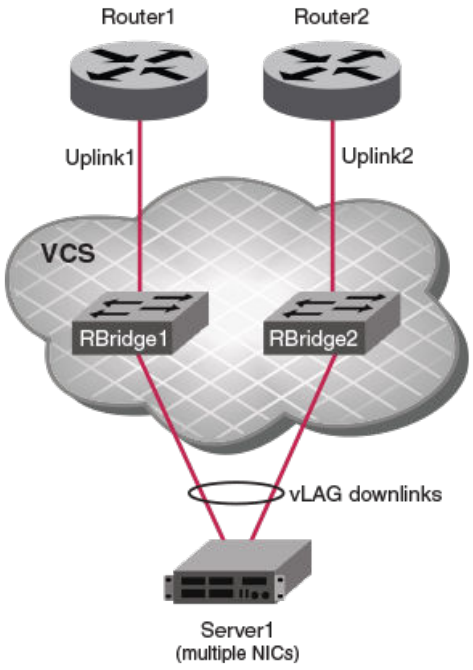
- In a VCS cluster, only local RBridge uplinks and downlinks are supported for LST; remote ports (even within the same VCS) are not supported.
- In a VCS cluster, only Ethernet FlexPorts are supported for LST. Fibre channel FlexPorts are not supported.

Configuring LST on a VCS cluster

Use this procedure to configure LST on a topology that includes a VCS but no independent RBridges.

In this procedure, the downlink from the VCS is a vLAG and the uplink is a physical port. You can modify this procedure for related topologies.

FIGURE 44 VCS cluster for LST implementation



1. Log in to the VCS principal RBridge.
2. Enter **configure** to access global configuration mode.

```
device# configure
```

3. Enter the **interface** command to access the downlink interface.

```
device(config)# interface port-channel 1
```

4. For each uplink interface that you want to track, enter the **track interface** command.

```
device(config-port-channel-1)# track interface ethernet 1/0/10
device(config-port-channel-1)# track interface ethernet 1/0/11
```

5. To modify the default behavior under multiple uplinks on VCS RBridges (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(config-port-channel-1)# track min-link 1
```

   In this case, the downlink vLAG member port on RBridge1 shuts down only if Uplink1 goes down. Similarly, the downlink vLAG member port on RBridge2 shuts down only if Uplink2 goes down.

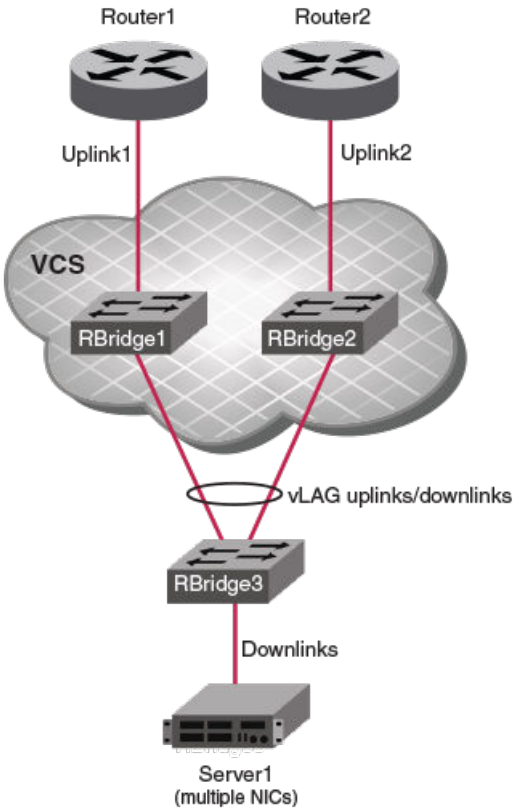
6. Enter the **track enable** command to enable tracking.

```
device(config-port-channel-1)# track enable
```

Configuring LST on a VCS cluster and an independent RBridge

Use this procedure to configure LST on a topology that includes a VCS and an independent RBridge.

FIGURE 45 VCS cluster and independent RBridge for LST implementation



1. Log in to the independent RBridge and enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 3/0/1
```

3. Enter the **track interface** command for the VCS principal RBridge.

```
device(conf-if-te-3/0/1)# track interface port-channel 30
```

4. To enable tracking of the VCS from the independent RBridge, enter the **track enable** command.

```
device(conf-if-te-3/0/1)# track enable
```

5. Log out of the independent RBridge.

```
device(conf-if-te-3/0/1)# end
device# exit
```

6. Log in to the VCS principal RBridge and enter **configure** to access global configuration mode.

```
device# configure
```

7. Enter the **interface** command to access the downlink interface.

```
device(config)# interface port-channel 1
```

8. For each uplink interface that you want to track, enter the **track interface** command.

```
device(config-port-channel-1)# track interface ethernet 1/0/10
device(config-port-channel-1)# track interface ethernet 2/0/11
```

9. To modify the default behavior under multiple uplinks on VCS RBridges (the downlink shuts down if any of the uplinks go down), enter a **track min-link** command.

```
device(config-port-channel-1)# track min-link 1
```

   In this case, the downlink vLAG member port on RBridge1 shuts down only if Uplink1 goes down. Similarly, the downlink vLAG member port on RBridge2 shuts down only if Uplink2 goes down.

10. Enter the **track enable** command to enable tracking.

```
device(config-port-channel-1)# track enable
```

Disabling LST

Use this procedure to disable link-state tracking on an interface, either partially or completely.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. Enter the **interface** command to access the downlink interface.

```
device(config)# interface tengigabitethernet 1/0/1
```

3. To disable uplink tracking, perform one of the following:

- To remove a specific uplink from tracking, enter the **no track interface** command

```
device(conf-if-te-1/0/1)# no track interface ethernet 1/0/20
```

- To disable tracking of all uplinks from this interface, enter the **no track enable** command.

```
device(conf-if-te-1/0/1)# no track enable
```

LST show commands

There is a range of show commands that include link-state tracking (LST) information. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 52 LST Show commands in the Network OS Command Reference

Command	Description
show interface	Displays the detailed interface configuration and capabilities of all interfaces or a specific interface. This command also indicates if an interface was brought down by LST because its uplinks were down.
show running-config interface <*N*>**gigabitethernet**	Displays configuration information about one or all device interfaces of a specific capacity. Replace <*N*>**gigabitethernet** with the desired operand (for example, **tengigabitethernet**). This command also indicates on which downlinks LST is defined, their enablement status, and any tracked uplinks.
show track summary	Displays LST details for all interfaces on a device.

Resolving Repeated MAC-Moves

Overview of resolving repeated MAC-moves

Repeated MAC-moves, often caused by loops, overload control-plane resources. Resolving MAC-moves usually solves the problem.

We use the term *repeated MAC-move*, or *MAC move*, to refer to the following sequence:

1. A frame is received on an interface and its MAC address is recorded in the MAC address table.
2. A frame with the identical MAC is received on a different interface on the same switch or VCS.

Following each MAC move, the control plane is interrupted to update the MAC address table with the most recent interface on which such a MAC address was seen. Resolution of repeated MAC-moves is necessary to avoid excessive demand on control plane resources.

Repeated MAC-moves can also lead to MAC table inconsistency across nodes in the cluster. Because the MACs learned on each RBridge are synchronized with the other RBridges in the cluster, the table inconsistences may lead to traffic flooding or black-holing. There is also the danger that even on a given RBridge, there may be inconsistencies among MAC addresses on the management module (MM), line cards (LCs), and the driver.

The following issues can result in repeated MAC-moves:

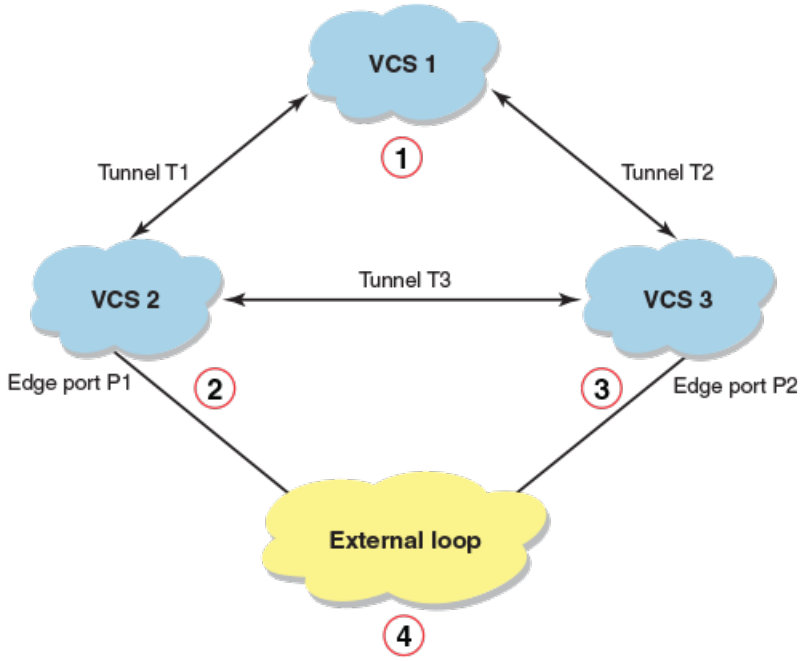
- Network loops
- Server-side flapping
- Other hardware or software issues

MAC-move detection

MAC-move detection provides a mechanism to detect too-frequent MAC moves across interfaces. This feature supports frequent MAC move detection globally at the VCS level, and also provides a mechanism to detect frequent MAC moves between interfaces across the VCS cluster in an overlay environment.

Consider the VCS overlay topology in the following figure.

FIGURE 46 MAC-move detection in a VCS overlay topology



1. MAC-move will be detected between tunnel T1 and tunnel T2.
2. MAC-move will be detected between edge port P1 and tunnel T3.
3. MAC-move will be detected between edge port P2 and tunnel T3.
4. MAC address will keep moving between edge port P1 in VCS 2 and edge port P2 in VCS 3.

The edge ports P1 in VCS 2 and P2 in VCS 3 will be shut down. No action will be taken on VCS 1. The traffic for these ports is black-holed. Consequently, the user must unshut one of the ports so that traffic resumes and a loop is prevented.

MAC-move detection is disabled by default.

If you enable this feature, by means of the **mac-address-table mac-move detect** command, you have the option of specifying the MAC-move threshold (the maximum number of moves allowed within any 10-second window). This is done by means of the **mac-address-table mac-move limit** command. An interface that exceeds the threshold is automatically shut down, and a "Repeated MAC-move" RASLog message is generated. To restore such an interface, you must enter the **no shutdown** command on that interface.

NOTE
The interface must not be a tunnel interface. It must be an edge port.

NOTE
For details of MAC-move resolution that does not require entering a restoration command, see "Administering Edge-Loop Detection."

MAC consistency-check

MAC consistency check provides a mechanism to detect MAC-address inconsistencies across learning modules and the driver in an RBridge or across RBridges in a VCS.

MAC consistency-check is enabled by default. You have the option of specifying the consistency-check interval (the number of seconds between consistency checks). If this feature detects MAC-address inconsistency, it corrects the inconsistent MAC addresses.

Configuring MAC-move detection for an entire VCS cluster

MAC-move detection is disabled by default. You can enable MAC-move detection by means of the **mac-address-table mac-move detect** command, and specify a threshold for repeated MAC-move detection by means of the **mac-address-table mac-move limit** command. The default threshold for repeated-MAC-move detection is 20 moves within any 10-second window.

This use of the **mac-address-table mac-move detect** command detects moves on the entire VCS cluster. Refer to "MAC-move detection" in this chapter.

1. Enter the **configure terminal** command to access global configuration mode.

```
device# configure terminal
```

2. To enable MAC-move detection, enter the **mac-address-table mac-move detect** command.

```
device(config)# mac-address-table mac-move detect
```

3. To modify the default MAC-move threshold, enter the **mac-address-table mac-move limit** command.

```
device(config)# mac-address-table mac-move limit 10
```

4. To restore the default MAC-move threshold of 20 moves, enter the **no mac-address-table mac-move limit** command.

```
device(config)# no mac-address-table mac-move limit
```

5. To disable MAC-move detection, enter the **no mac-address-table mac-move detect** command.

```
device(config)# no mac-address-table mac-move detect
```

6. You can use the **show mac-address-table mac-move** command or the **show ip interface brief** command to confirm the configuration, as in the following examples, respectively.

```
device# show mac-address-table mac-move
Mac Move detect :      Disabled
Threshold:             10

device# show ip interface brief
Interface                  IP-Address     Status    Protocol
============================================================
Port-channel 1             unassigned        up       up
TenGigabitEthernet 1/0/1   unassigned        up       up
TenGigabitEthernet 1/0/2   unassigned  admin-down     down  <-- Shut down as a result of exceeded
threshold
```

Configuring MAC consistency check

MAC consistency check examines MAC-address consistency across VCS RBridges. If needed, this feature resynchronizes the MAC addresses.

NOTE
By default, MAC consistency check is enabled.

1. Enter **configure** to access global configuration mode.

```
device# configure
```

2. To modify the consistency-check interval (default: 300 seconds), enter the **mac-address-table consistency-check interval** command.

```
device(config)# mac-address-table consistency-check interval 500
```

3. To restore the default consistency-check interval of 300 seconds, enter the **no mac-address-table consistency-check interval** command.

```
device(config)# no mac-address-table consistency-check interval
```

4. To disable MAC consistency check, enter the **mac-address-table consistency-check suppress** command.

```
device(config)# mac-address-table consistency-check suppress
```

5. To restore enablement of MAC consistency check, enter the **no mac-address-table consistency-check suppress** command.

```
device(config)# no mac-address-table consistency-check suppress
```

MAC-move show commands

There are several **show** commands that display repeated-MAC-move information. They are documented in the *Network OS Command Reference*, and listed here with descriptions.

TABLE 53 MAC-move show commands

Command	Description
show interface *interface_id*	Displays MAC-move information on a per-interface basis.
show ip/ipv6 interface brief	Indicates whether an interface was shut down due to repeated MAC moves.
show mac-address-table consistency-check	Displays the operational data for MAC address-table consistency check.
show mac-address-table mac-move	Displays MAC-move threshold and enable/disable information.