Ethernet Switching Features on the QFX Series

Release
13.2X52

Modified: 2015-10-13

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Ethernet Switching Features on the QFX Series
13.2X52

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- QFabric System

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xvii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xvii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric** ***metric*****>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [** ***community-ids*** **]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at http://www.juniper.net/techpubs/feedback/.
- E-mail—Send your comments to **techpubs-comments@juniper.net**. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Software Features Overview

Overview of Layer 2 Networking

Layer 2, also known as the Data Link Layer, is the second level in the seven-layer OSI reference model for network protocol design. Layer 2 is equivalent to the link layer (the lowest layer) in the TCP/IP network model. Layer2 is the network layer used to transfer data between adjacent network nodes in a wide area network or between nodes on the same local area network.

A *frame* is a protocol data unit, the smallest unit of bits on a Layer 2 network. Frames are transmitted to and received from devices on the same local area network (LAN). Unilke bits, frames have a defined structure and can be used for error detection, control plane activities and so forth. Not all frames carry user data. The network uses some frames to control the data link itself..

At Layer 2, *unicast* refers to sending frames from one node to a single other node, whereas *multicast* denotes sending traffic from one node to multiple nodes, and *broadcasting* refers to the transmission of frames to all nodes in a network. A *broadcast domain* is a logical division of a network in which all nodes of that network can be reached at Layer 2 by a broadcast.

Segments of a LAN can be linked at the frame level using *bridges*. Bridging creates separate broadcast domains on the LAN, creating VLANs, which are independent logical networks that group together related devices into separate network segments. The grouping of devices on a VLAN is independent of where the devices are physically located in the LAN. Without bridging and VLANs, all devices on the Ethernet LAN are in a single broadcast domain, and all the devices detect all the packets on the LAN.

Forwarding is the relaying of packets from one network segment to another by nodes in the network. On a VLAN, a frame whose origin and destination are in the same VLAN are forwarded only within the local VLAN. A network segment is a portion of a computer network wherein every device communicates using the same physical layer.

Layer 2 contains two sublayers:

- Logical link control (LLC) sublayer, which is responsible for managing communications links and handling frame traffic.
- Media access control (MAC) sublayer, which governs protocol access to the physical network medium. By using the MAC addresses that are assigned to all ports on a switch, multiple devices on the same physical link can uniquely identify one another.

  The ports, or interfaces, on a switch operate in either access mode, tagged-access, or trunk mode:

  - *Access mode* ports connect to a network device such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. The port itself belongs to a single VLAN. The frames transmitted over an access interface are normal Ethernet frames. By default, all ports on a switch are in access mode.
  - *Tagged-Access mode* ports connect to a network device such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. The port itself belongs to a single VLAN. The frames transmitted over an access interface are normal Ethernet frames. By default, all ports on a switch are in access mode. Tagged-access mode accommodates cloud computing, specifically scenarios including virtual machines or virtual computers. Because several virtual computers can be included on one physical server, the packets generated by one server can contain an aggregation of VLAN packets from different virtual machines on that server. To accommodate this situation, tagged-access mode reflects packets back to the physical server on the same downstream port when the destination address of the packet was learned on that downstream port. Packets are also reflected back to the physical server on the downstream port when the destination has not yet been learned. Therefore, the third interface mode, tagged access, has some characteristics of access mode and some characteristics of trunk mode:
  - *Trunk mode* ports handle traffic for multiple VLANs, multiplexing the traffic for all those VLANs over the same physical connection. Trunk interfaces are generally used to interconnect switches to other devices or switches.

    With native VLAN configured, frames that do not carry VLAN tags are sent over the trunk interface. If you have a situation where packets pass from a device to a switch in access mode, and you want to then send those packets from the switch over a trunk port, use native VLAN mode. Configure the single VLAN on the switch's port (which is in access mode) as a native VLAN. The switch's trunk port will then treat those frames differently than the other tagged packets. For example, if a trunk port has three VLANs, 10, 20, and 30, assigned to it with VLAN 10 being the native VLAN, frames on VLAN 10 that leave the trunk port on the other end have no 802.1Q header (tag). There is another native VLAN option. You can have the switch add and remove tags for untagged packets. To do this, you first configure the single VLAN as a native VLAN on a port attached to a device on the edge. Then, assign a VLAN ID tag to the single native VLAN on the port connected to a device. Last, add the VLAN ID to the trunk port. Now, when the switch receives the untagged packet, it adds the ID you specified and sends and receives the tagged packets on the trunk port configured to accept that VLAN.

Including the sublayers, Layer 2 on the QFX Series supports the following functionality:

- Unicast, multicast, and broadcast traffic.
- Bridging.
- VLAN 802.1Q—Also known as *VLAN tagging*, this protocol allows multiple bridged networks to transparently share the same physical network link by adding VLAN tags to an Ethernet frame.
- Extension of Layer 2 VLANs across multiple switches using Spanning Tree Protocol (STP) prevents looping across the network.
- *MAC learning*, including per-VLAN MAC learning and Layer 2 learning suppression–This process obtains the MAC addresses of all the nodes on a network
- Link aggregation—This process groups of Ethernet interfaces at the physical layer to form a single link layer interface, also known as a *link aggregation group (LAG)* or LAG bundle
- Storm control on the physical port for unicast, multicast, and broadcast
- STP support, including 802.1d, RSTP, MSTP, and Root Guard

Related Documentation

- Understanding Bridging and VLANs on page 5
- *Understanding Bridging*

Understanding Bridging and VLANs

Network switches use Layer 2 bridging protocols to discover the topology of their LAN and to forward traffic toward destinations on the LAN. This topic explains the following concepts regarding bridging and VLANs:

History of VLANs

Ethernet LANs were originally designed for small, simple networks that primarily carried text. However, over time, the type of data carried by LANs grew to include voice, graphics,

and video. This more complex data, when combined with the ever-increasing speed of transmission, eventually became too much of a load for the original Ethernet LAN design. Multiple packet collisions were significantly slowing down the larger LANs.

The IEEE 802.1D-2004 standard helped evolve Ethernet LANs to cope with the higher data and transmission requirements by defining the concept of *transparent bridging* (generally called simply *bridging*). Bridging divides a single physical LAN (now called a single *broadcast domain*) into two or more virtual LANs, or VLANs. Each *VLAN* is a collection of some of the LAN nodes grouped together to form individual broadcast domains.

When VLANs are grouped logically by function or organization, a significant percentage of data traffic stays within the VLAN. This relieves the load on the LAN because all traffic no longer has to be forwarded to all nodes on the LAN. A VLAN first transmits packets within the VLAN, thereby reducing the number of packets transmitted on the entire LAN. Because packets whose origin and destination are in the same VLAN are forwarded only within the local VLAN, packets that are not destined for the local VLAN are the only ones forwarded to other broadcast domains. This way, bridging and VLANs limit the amount of traffic flowing across the entire LAN by reducing the possible number of collisions and packet retransmissions within VLANs and on the LAN as a whole.

How Bridging of VLAN Traffic Works

Because the objective of the IEEE 802.1D-2004 standard was to reduce traffic and therefore reduce potential transmission collisions for Ethernet, a system was implemented to reuse information. Instead of having a switch go through a location process every time a frame is sent to a node, the transparent bridging protocol allows a switch to record the location of known nodes. When packets are sent to nodes, those destination node locations are stored in address-lookup tables called *Ethernet switching tables*. Before sending a packet, a switch using bridging first consults the switching tables to see if that node has already been located. If the location of a node is known, the frame is sent directly to that node.

Transparent bridging uses five mechanisms to create and maintain Ethernet switching tables on the switch:

- Learning
- Forwarding
- Flooding
- Filtering
- Aging

The key bridging mechanism used by LANs and VLANs is *learning*. When a switch is first connected to an Ethernet LAN or VLAN, it has no information about other nodes on the network. As packets are sent, the switch learns the embedded MAC addresses of the sending nodes and stores them in the Ethernet switching table, along with two other pieces of information—the interface (or port) on which the traffic was received on the destination node and the time the address was learned.

Learning allows switches to then do *forwarding*. By consulting the Ethernet switching table to see whether the table already contains the frame's destination MAC address, switches save time and resources when forwarding packets to the known MAC addresses. If the Ethernet switching table does not contain an entry for an address, the switch uses flooding to learn that address.

Flooding finds a particular destination MAC address without using the Ethernet switching table. When traffic originates on the switch and the Ethernet switching table does not yet contain the destination MAC address, the switch first floods the traffic to all other interfaces within the VLAN. When the destination node receives the flooded traffic, it can send an acknowledgment packet back to the switch, allowing it to learn the MAC address of the node and add the address to its Ethernet switching table.

Filtering, the fourth bridging mechanism, is how broadcast traffic is limited to the local VLAN whenever possible. As the number of entries in the Ethernet switching table grows, the switch pieces together an increasingly complete picture of the VLAN and the larger LAN—it learns which nodes are in the local VLAN and which are on other network segments. The switch uses this information to filter traffic. Specifically, for traffic whose source and destination MAC addresses are in the local VLAN, filtering prevents the switch from forwarding this traffic to other network segments.

To keep entries in the Ethernet switching table current, the switch uses a fifth bridging mechanism, *aging*. Aging is the reason that the Ethernet switching table entries include timestamps. Each time the switch detects traffic from a MAC address, it updates the timestamp. A timer on the switch periodically checks the timestamp, and if it is older than a user-configured value, the switch removes the node's MAC address from the Ethernet switching table. This aging process eventually flushes unavailable network nodes out of the Ethernet switching table.

Packets Are Either Tagged or Untagged

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q ID. The number of available VLANs and VLAN IDs are listed below:

- On a switch running ELS software, you can configure 4093 VLANs.
- On a switch running non-ELS software, you can configure 4091 VLANs.

Ethernet packets include a tag protocol identifier (TPID) EtherType field, which identifies the protocol being transported. When a device within a VLAN generates a packet, this field includes a value of 0x8100, which indicates that the packet is a VLAN-tagged packet. The packet also has a VLAN ID field that includes the unique 802.1Q ID, which identifies the VLAN to which the packet belongs.

In addition to the TPID EtherType value of 0x8100, switches that run Junos OS that does not support the Enhanced Layer 2 Software (ELS) configuration style also support values of 0x88a8 (Provider Bridging and Shortest Path Bridging) and 0x9100 (Q-inQ).

For a simple network that has only a single VLAN, all packets include a default 802.1Q tag, which is the only VLAN membership that does not mark the packet as tagged. These packets are untagged packets.

Switch Interface Modes—Access, Trunk, or Tagged Access

Ports, or interfaces, on a switch operate in one of three modes:

- Access mode
- Trunk mode
- Tagged-access mode

Access Mode

An interface in access mode connects a switch to a single network device, such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. Access interfaces accept only untagged packets.

By default, when you boot a switch that runs Junos OS that does not support ELS and use the factory default configuration, or when you boot such a switch and do not explicitly configure a port mode, all interfaces on the switch are in access mode and accept only untagged packets from the VLAN named **default**. You can optionally configure another VLAN and use that VLAN instead of **default**.

On a switch that runs Junos OS that supports ELS, the VLAN named **default** is not supported. Therefore, on such switches, you must explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist. After you assign an interface to a VLAN, the interface functions in access mode.

For switches that run either type of software, you can also configure a trunk port or interface to accept untagged packets from a user-configured VLAN. For details about this concept (native VLAN), see "Trunk Mode and Native VLAN" on page 9.

Trunk Mode

Trunk mode interfaces are generally used to connect switches to one another. Traffic sent between switches can then consist of packets from multiple VLANs, with those packets multiplexed so that they can be sent over the same physical connection. Trunk interfaces usually accept only tagged packets and use the VLAN ID tag to determine both the packets' VLAN origin and VLAN destination.

On a switch that runs software that does not support ELS, an untagged packet is not recognized on a trunk port unless you configure additional settings on that port.

On a switch that runs Junos OS that supports ELS, a trunk port recognizes untagged control packets for protocols such as the Link Aggregation Control Protocol (LACP) and the Link Layer Discovery Protocol (LLDP). However, the trunk port does not recognize untagged data packets unless you configure additional settings on that port.

In the rare case where you want untagged packets to be recognized by a trunk port on switches that run either type of software, you must configure the single VLAN on a trunk port as a *native VLAN*. For more information about native VLANs, see "Trunk Mode and Native VLAN" on page 9.

Trunk Mode and Native VLAN

On a switch that runs Junos OS that does not support ELS, a trunk port does not recognize packets that do not include VLAN tags, which are also known an untagged packets. On a switch that runs Junos OS that supports ELS, a trunk port recognizes untagged control packets, but it does not recognize untagged data packets. With native VLAN configured, untagged packets that a trunk port normally does not recognize are sent over the trunk interface. In a situation where packets pass from a device, such as an IP phone or printer, to a switch in access mode, and you want those packets sent from the switch over a trunk port, use native VLAN mode. Create a native VLAN by configuring a VLAN ID for it, and specify that the trunk port is a member of the native VLAN.

The switch's trunk port will then treat those packets differently than the other tagged packets. For example, if a trunk port has three VLANs, 10, 20, and 30, assigned to it with VLAN 10 being the native VLAN, packets on VLAN 10 that leave the trunk port on the other end have no 802.1Q header (tag).

There is another native VLAN option for switches that do not support ELS. You can have the switch add and remove tags for untagged packets. To do this, you first configure the single VLAN as a native VLAN on a port attached to a device on the edge. Then, assign a VLAN ID tag to the single native VLAN on the port connected to a device. Last, add the VLAN ID to the trunk port. Now, when the switch receives the untagged packet, it adds the ID you specified and sends and receives the tagged packets on the trunk port configured to accept that VLAN.

Tagged-Access Mode

Only switches that run Junos OS that does not use the ELS configuration style support tagged-access mode. Tagged-access mode accommodates cloud computing, specifically scenarios including virtual machines or virtual computers. Because several virtual computers can be included on one physical server, the packets generated by one server can contain an aggregation of VLAN packets from different virtual machines on that server. To accommodate this situation, tagged-access mode reflects packets back to the physical server on the same downstream port when the destination address of the packet was learned on that downstream port. Packets are also reflected back to the physical server on the downstream port when the destination has not yet been learned. Therefore, the third interface mode, tagged access, has some characteristics of access mode and some characteristics of trunk mode:

- Like access mode, tagged-access mode connects the switch to an access layer device. Unlike access mode, tagged-access mode is capable of accepting VLAN tagged packets.
- Like trunk mode, tagged-access mode accepts VLAN tagged packets from multiple VLANs. Unlike trunk port interfaces, which are connected at the core/distribution layer, tagged-access port interfaces connect devices at the access layer.

  Like trunk mode, tagged-access mode also supports native VLAN.

NOTE: **Control packets are never reflected back on the downstream port.**

Additional Advantages of Using VLANs

In addition to reducing traffic and thereby speeding up the network, VLANs have the following advantages:

- VLANs provide segmentation services traditionally provided by routers in LAN configurations, thereby reducing hardware equipment costs.
- Packets coupled to a VLAN can be reliably identified and sorted into different domains. You can contain broadcasts within parts of the network, thereby freeing up network resources. For example, when a DHCP server is plugged into a switch and starts broadcasting its presence, you can prevent some hosts from accessing it by using VLANs to split up the network.
- For security issues, VLANs provide granular control of the network because each VLAN is identified by a single IP subnetwork. All packets passing in and out of a VLAN are consistently tagged with the VLAN ID of that VLAN, thereby providing easy identification, because a VLAN ID on a packet cannot be altered. (For a switch that runs Junos OS that does not support ELS, we recommend that you avoid using 1 as a VLAN ID, because that ID is a default value.)
- VLANs react quickly to host relocation—this is also due to the persistent VLAN tag on packets.
- On an Ethernet LAN, all network nodes must be physically connected to the same network. In VLANs, the physical location of nodes is not important—you can group network devices in any way that makes sense for your organization, such as by department or business function, types of network nodes, or physical location.

Maximum VLANs and VLAN Members Per Switch

The number of VLANs supported per switch varies for each switch. Use the configuration-mode command **set vlans *vlan-name* vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because you have to assign a specific ID number when you create a VLAN—you could overwrite one of the numbers, but you cannot exceed the limit.

You can, however, exceed the recommended VLAN member maximum for a switch.

On a switch that runs Junos OS that does not support the ELS configuration style, the maximum number of VLAN members allowed on the switch is eight times the maximum number of VLANs that the switch supports (vmember limit = vlan max * 8). If the configuration of the switch exceeds the recommended VLAN member maximum, a warning message appears when you commit the configuration. If you commit the configuration despite the warning, the commit succeeds, but there is a risk of the Ethernet switching process (eswd) failing as a result of memory allocation failure.

On a switch that runs Junos OS that supports ELS, the maximum number of VLAN members allowed on the switch is 24 times the maximum number of VLANs that the switch supports (vmember limit = vlan max * 24). If the configuration of the switch exceeds the recommended VLAN member maximum, a warning message appears in the system log (syslog).

A QFabric system supports up to 131,008 VLAN members (vmembers) on a single network node group, server node group, or redundant server node group. The number of vmembers is calculated by multiplying the maximum number of VLANs by 32.

For example, to calculate how many interfaces are required to support 4,000 VLANs, divide the maximum number of vmembers (128,000) by the number of configured VLANs (4,000). In this case, 32 interfaces are required.

On network Node groups and server Node groups, you can configure link aggregation groups (LAGs) across multiple interfaces. Each LAG and VLAN combination is considered a vmember.

A Default VLAN Is Configured on Most Switches

Some switches that run Junos OS that do not support the ELS configuration style are preconfigured with a VLAN named **default** that does not tag packets and operates only with untagged packets. On these switches, each interface already belongs to the VLAN named **default** and all traffic uses this VLAN until you configure more VLANs and assign traffic to those VLANs.

NOTE: When a Juniper Networks QFX3500 or QFX3600 switch is interconnected with other switches in a Virtual Chassis configuration, each individual switch that is included as a member of the configuration is identified with a member ID. The member ID functions as an FPC slot number. When you are configuring interfaces for a Virtual Chassis configuration, you specify the appropriate member ID (0 through 9) as the slot element of the interface name. The default factory settings for a Virtual Chassis configuration include FPC 0 as a member of the default VLAN because FPC 0 is configured as part of the ethernet-switching family. In order to include FPC 1 through FPC 9 in the default VLAN, add the ethernet-switching family to the configurations for those interfaces.

Assigning Traffic to VLANs

You can assign traffic on any switch to a particular VLAN by referencing either the interface port of the traffic or the MAC addresses of devices sending traffic.

Assign VLAN Traffic According to the Interface Port Source

This method is most commonly used to assign traffic to VLANs. In this case, you specify that all traffic received on a particular switch interface is assigned to a specific VLAN. You configure this VLAN assignment when you configure the switch, by using either the VLAN number (called a VLAN ID) or by using the VLAN name, which the switch then translates into a numeric VLAN ID. This method is referred to simply as creating a VLAN because it is the most commonly used method.

Assign VLAN Traffic According to the Source MAC Address

In this case, all traffic received from a specific MAC address is forwarded to a specific egress interface (next hop) on the switch. MAC-based VLANs are either static (named MAC addresses configured one at a time) or dynamic (configured using a RADIUS server).

To configure a static MAC-based VLAN on a switch that supports ELS, see *Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)*.To configure a static MAC-based VLAN on a switch that does not support ELS, see *Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)*.

Forwarding VLAN Traffic

To pass traffic within a VLAN, the switch uses Layer 2 forwarding protocols, including IEEE 802.1Q spanning-tree protocols.

To pass traffic between two VLANs, the switch uses standard Layer 3 routing protocols, such as static routing, OSPF, and RIP. The same interfaces that support Layer 2 bridging protocols also support Layer 3 routing protocols, providing multilayer switching.

To pass traffic from a single device on an access port to a switch and then pass those packets on a trunk port, use the native mode configuration previously discussed under "Trunk Mode" on page 8.

VLANs Communicate with Integrated Routing and Bridging Interfaces or Routed VLAN Interfaces

Traditionally, switches sent traffic to hosts that were part of the same broadcast domain (VLAN) but routers were needed to route traffic from one broadcast domain to another. Also, only routers performed other Layer 3 functions such as traffic engineering.

Switches that run Junos OS that supports the ELS configuration style perform inter-VLAN routing functions using an integrated routing and bridging (IRB) interface named irb, while switches that run Junos OS that does not support ELS perform these functions using a routed VLAN interface (RVI) named vlan. These interfaces detect both MAC addresses and IP addresses and route data to Layer 3 interfaces, thereby frequently eliminating the need to have both a switch and a router.

NOTE:

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Understanding FCoE*
- *Interfaces Overview*

Understanding Unicast

Unicasting is the act of sending data from one node of the network to another. In contrast, multicast transmissions send traffic from one data node to multiple other data nodes.

Unknown unicast traffic consists of unicast frames with unknown destination MAC addresses. By default, the switch floods these unicast frames that are traveling in a VLAN to all interfaces that are members of the VLAN. Forwarding this type of traffic to interfaces on the switch can trigger a security issue. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service. This is known as a traffic storm.

To prevent a storm, you can disable the flooding of unknown unicast packets to all interfaces by configuring one VLAN or all VLANs to forward any unknown unicast traffic to a specific trunk interface. (This channels the unknown unicast traffic to a single interface.)

Related Documentation

- Overview of Layer 2 Networking on page 3
- Understanding Bridging and VLANs on page 5

Introduction to the Media Access Control (MAC) Layer 2 Sublayer

This topic provides an introduction to the MAC sublayer of the data link layer (Layer 2).

In Layer 2 of a network, the Media Access Control (MAC) sublayer provides addressing and channel access control mechanisms that enable several terminals or network nodes to communicate in a network.

The MAC sublayer acts as an interface between the logical link control (LLC) Ethernet sublayer and Layer 1 (the physical layer). The MAC sublayer emulates a full-duplex logical communication channel in a multipoint network. This channel may provide unicast, multicast, or broadcast communication service. The MAC sublayer uses MAC protocols to prevent collisions.

In Layer 2, multiple devices on the same physical link can uniquely identify one another at the data link layer, by using the MAC addresses that are assigned to all ports on a switch. A MAC algorithm accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC address.

A MAC address is a 12-digit hexadecimal number (48 bits in long). MAC addresses are usually written in one of these formats:

- MM:MM:MM:SS:SS:SS
- MM-MM-MM-SS-SS-SS

The first half of a MAC address contains the ID number of the adapter manufacturer. These IDs are regulated by an Internet standards body. The second half of a MAC address represents the serial number assigned to the adapter by the manufacturer.

Contrast MAC addressing, which works at Layer 2, with IP addressing, which runs at Layer 3 (networking and routing). One way to remember the difference is that the MAC addresses apply to a physical or virtual node, whereas IP addresses apply to the software implementation of that node. MAC addresses are typically fixed on a per-node basis, whereas IP addresses change when the node moves from one part of the network to another.

IP networks maintain a mapping between the IP and MAC addresses of a node using the Address Resolution Protocol (ARP) table. DHCP also typically uses MAC addresses when assigning IP addresses to nodes.

Related Documentation

- Overview of Layer 2 Networking on page 3
- Understanding MAC Learning on page 26

Understanding Layer 2 Broadcasting

In a Layer 2 network, *broadcasting* refers to sending traffic to all nodes on a network.

Layer 2 broadcast traffic stays within a local area network (LAN) boundary; known as the *broadcast domain*. Layer 2 broadcast traffic is sent to the broadcast domain using a MAC address of FF:FF:FF:FF:FF:FF. Every device in the broadcast domain recognizes this MAC address and passes the broadcast traffic on to other devices in the broadcast domain, if applicable. Broadcasting can be compared to unicasting (sending traffic to a single node) or multicasting (delivering traffic to a group of nodes simultaneously).

Layer 3 broadcast traffic, however, is sent to all devices in a network using a broadcast network address. For example, if your network address is 192.0.0.0, the broadcast network address is 192.255.255.255. In this case, only devices that belong to the 192.0.0.0 network receive the Layer 3 broadcast traffic. Devices that do not belong to this network drop the traffic.

Broadcasting is used in the following situations:

- Address Resolution Protocol (ARP) uses broadcasting to map MAC addresses to IP addresses. ARP dynamically binds the IP address (the logical address) to the correct MAC address. Before IP unicast packets can be sent, ARP discovers the MAC address used by the Ethernet interface where the IP address is configured.
- Dynamic Host Configuration Protocol (DHCP) uses broadcasting to dynamically assign IP addresses to hosts on a network segment or subnet.
- Routing protocols use broadcasting to advertise routes.

Excessive broadcast traffic can sometimes create a broadcast storm. A broadcast storm occurs when messages are broadcast on a network and each message prompts a receiving node to respond by broadcasting its own messages on the network. This, in turn, prompts further responses that create a snowball effect. The LAN is suddenly flooded with packets, creating unnecessary traffic that leads to poor network performance or even a complete loss of network service.

Related Documentation

- Overview of Layer 2 Networking on page 3
- *Understanding Storm Control*
- *Understanding Bridging*
- Understanding Bridging and VLANs on page 5

CHAPTER 2

Bridging and VLANs

Understanding Bridging and VLANs

Network switches use Layer 2 bridging protocols to discover the topology of their LAN and to forward traffic toward destinations on the LAN. This topic explains the following concepts regarding bridging and VLANs:

History of VLANs

Ethernet LANs were originally designed for small, simple networks that primarily carried text. However, over time, the type of data carried by LANs grew to include voice, graphics, and video. This more complex data, when combined with the ever-increasing speed of transmission, eventually became too much of a load for the original Ethernet LAN design. Multiple packet collisions were significantly slowing down the larger LANs.

The IEEE 802.1D-2004 standard helped evolve Ethernet LANs to cope with the higher data and transmission requirements by defining the concept of *transparent bridging* (generally called simply *bridging*). Bridging divides a single physical LAN (now called a single *broadcast domain*) into two or more virtual LANs, or VLANs. Each *VLAN* is a collection of some of the LAN nodes grouped together to form individual broadcast domains.

When VLANs are grouped logically by function or organization, a significant percentage of data traffic stays within the VLAN. This relieves the load on the LAN because all traffic no longer has to be forwarded to all nodes on the LAN. A VLAN first transmits packets within the VLAN, thereby reducing the number of packets transmitted on the entire LAN. Because packets whose origin and destination are in the same VLAN are forwarded only within the local VLAN, packets that are not destined for the local VLAN are the only ones forwarded to other broadcast domains. This way, bridging and VLANs limit the amount of traffic flowing across the entire LAN by reducing the possible number of collisions and packet retransmissions within VLANs and on the LAN as a whole.

How Bridging of VLAN Traffic Works

Because the objective of the IEEE 802.1D-2004 standard was to reduce traffic and therefore reduce potential transmission collisions for Ethernet, a system was implemented to reuse information. Instead of having a switch go through a location process every time a frame is sent to a node, the transparent bridging protocol allows a switch to record the location of known nodes. When packets are sent to nodes, those destination node locations are stored in address-lookup tables called *Ethernet switching tables*. Before sending a packet, a switch using bridging first consults the switching tables to see if that node has already been located. If the location of a node is known, the frame is sent directly to that node.

Transparent bridging uses five mechanisms to create and maintain Ethernet switching tables on the switch:

- Learning
- Forwarding
- Flooding
- Filtering
- Aging

The key bridging mechanism used by LANs and VLANs is *learning*. When a switch is first connected to an Ethernet LAN or VLAN, it has no information about other nodes on the

network. As packets are sent, the switch learns the embedded MAC addresses of the sending nodes and stores them in the Ethernet switching table, along with two other pieces of information—the interface (or port) on which the traffic was received on the destination node and the time the address was learned.

Learning allows switches to then do *forwarding*. By consulting the Ethernet switching table to see whether the table already contains the frame's destination MAC address, switches save time and resources when forwarding packets to the known MAC addresses. If the Ethernet switching table does not contain an entry for an address, the switch uses flooding to learn that address.

Flooding finds a particular destination MAC address without using the Ethernet switching table. When traffic originates on the switch and the Ethernet switching table does not yet contain the destination MAC address, the switch first floods the traffic to all other interfaces within the VLAN. When the destination node receives the flooded traffic, it can send an acknowledgment packet back to the switch, allowing it to learn the MAC address of the node and add the address to its Ethernet switching table.

Filtering, the fourth bridging mechanism, is how broadcast traffic is limited to the local VLAN whenever possible. As the number of entries in the Ethernet switching table grows, the switch pieces together an increasingly complete picture of the VLAN and the larger LAN—it learns which nodes are in the local VLAN and which are on other network segments. The switch uses this information to filter traffic. Specifically, for traffic whose source and destination MAC addresses are in the local VLAN, filtering prevents the switch from forwarding this traffic to other network segments.

To keep entries in the Ethernet switching table current, the switch uses a fifth bridging mechanism, *aging*. Aging is the reason that the Ethernet switching table entries include timestamps. Each time the switch detects traffic from a MAC address, it updates the timestamp. A timer on the switch periodically checks the timestamp, and if it is older than a user-configured value, the switch removes the node's MAC address from the Ethernet switching table. This aging process eventually flushes unavailable network nodes out of the Ethernet switching table.

Packets Are Either Tagged or Untagged

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q ID. The number of available VLANs and VLAN IDs are listed below:

- On a switch running ELS software, you can configure 4093 VLANs.
- On a switch running non-ELS software, you can configure 4091 VLANs.

Ethernet packets include a tag protocol identifier (TPID) EtherType field, which identifies the protocol being transported. When a device within a VLAN generates a packet, this field includes a value of 0x8100, which indicates that the packet is a VLAN-tagged packet. The packet also has a VLAN ID field that includes the unique 802.1Q ID, which identifies the VLAN to which the packet belongs.

In addition to the TPID EtherType value of 0x8100, switches that run Junos OS that does not support the Enhanced Layer 2 Software (ELS) configuration style also support values of 0x88a8 (Provider Bridging and Shortest Path Bridging) and 0x9100 (Q-inQ).

For a simple network that has only a single VLAN, all packets include a default 802.1Q tag, which is the only VLAN membership that does not mark the packet as tagged. These packets are untagged packets.

Switch Interface Modes—Access, Trunk, or Tagged Access

Ports, or interfaces, on a switch operate in one of three modes:

- Access mode
- Trunk mode
- Tagged-access mode

Access Mode

An interface in access mode connects a switch to a single network device, such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. Access interfaces accept only untagged packets.

By default, when you boot a switch that runs Junos OS that does not support ELS and use the factory default configuration, or when you boot such a switch and do not explicitly configure a port mode, all interfaces on the switch are in access mode and accept only untagged packets from the VLAN named **default**. You can optionally configure another VLAN and use that VLAN instead of **default**.

On a switch that runs Junos OS that supports ELS, the VLAN named **default** is not supported. Therefore, on such switches, you must explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist. After you assign an interface to a VLAN, the interface functions in access mode.

For switches that run either type of software, you can also configure a trunk port or interface to accept untagged packets from a user-configured VLAN. For details about this concept (native VLAN), see "Trunk Mode and Native VLAN" on page 9.

Trunk Mode

Trunk mode interfaces are generally used to connect switches to one another. Traffic sent between switches can then consist of packets from multiple VLANs, with those packets multiplexed so that they can be sent over the same physical connection. Trunk interfaces usually accept only tagged packets and use the VLAN ID tag to determine both the packets' VLAN origin and VLAN destination.

On a switch that runs software that does not support ELS, an untagged packet is not recognized on a trunk port unless you configure additional settings on that port.

On a switch that runs Junos OS that supports ELS, a trunk port recognizes untagged control packets for protocols such as the Link Aggregation Control Protocol (LACP) and

the Link Layer Discovery Protocol (LLDP). However, the trunk port does not recognize untagged data packets unless you configure additional settings on that port.

In the rare case where you want untagged packets to be recognized by a trunk port on switches that run either type of software, you must configure the single VLAN on a trunk port as a *native VLAN*. For more information about native VLANs, see "Trunk Mode and Native VLAN" on page 9.

Trunk Mode and Native VLAN

On a switch that runs Junos OS that does not support ELS, a trunk port does not recognize packets that do not include VLAN tags, which are also known an untagged packets. On a switch that runs Junos OS that supports ELS, a trunk port recognizes untagged control packets, but it does not recognize untagged data packets. With native VLAN configured, untagged packets that a trunk port normally does not recognize are sent over the trunk interface. In a situation where packets pass from a device, such as an IP phone or printer, to a switch in access mode, and you want those packets sent from the switch over a trunk port, use native VLAN mode. Create a native VLAN by configuring a VLAN ID for it, and specify that the trunk port is a member of the native VLAN.

The switch's trunk port will then treat those packets differently than the other tagged packets. For example, if a trunk port has three VLANs, 10, 20, and 30, assigned to it with VLAN 10 being the native VLAN, packets on VLAN 10 that leave the trunk port on the other end have no 802.1Q header (tag).

There is another native VLAN option for switches that do not support ELS. You can have the switch add and remove tags for untagged packets. To do this, you first configure the single VLAN as a native VLAN on a port attached to a device on the edge. Then, assign a VLAN ID tag to the single native VLAN on the port connected to a device. Last, add the VLAN ID to the trunk port. Now, when the switch receives the untagged packet, it adds the ID you specified and sends and receives the tagged packets on the trunk port configured to accept that VLAN.

Tagged-Access Mode

Only switches that run Junos OS that does not use the ELS configuration style support tagged-access mode. Tagged-access mode accommodates cloud computing, specifically scenarios including virtual machines or virtual computers. Because several virtual computers can be included on one physical server, the packets generated by one server can contain an aggregation of VLAN packets from different virtual machines on that server. To accommodate this situation, tagged-access mode reflects packets back to the physical server on the same downstream port when the destination address of the packet was learned on that downstream port. Packets are also reflected back to the physical server on the downstream port when the destination has not yet been learned. Therefore, the third interface mode, tagged access, has some characteristics of access mode and some characteristics of trunk mode:

- Like access mode, tagged-access mode connects the switch to an access layer device. Unlike access mode, tagged-access mode is capable of accepting VLAN tagged packets.

- Like trunk mode, tagged-access mode accepts VLAN tagged packets from multiple VLANs. Unlike trunk port interfaces, which are connected at the core/distribution layer, tagged-access port interfaces connect devices at the access layer.

  Like trunk mode, tagged-access mode also supports native VLAN.

NOTE: **Control packets are never reflected back on the downstream port.**

Additional Advantages of Using VLANs

In addition to reducing traffic and thereby speeding up the network, VLANs have the following advantages:

- VLANs provide segmentation services traditionally provided by routers in LAN configurations, thereby reducing hardware equipment costs.
- Packets coupled to a VLAN can be reliably identified and sorted into different domains. You can contain broadcasts within parts of the network, thereby freeing up network resources. For example, when a DHCP server is plugged into a switch and starts broadcasting its presence, you can prevent some hosts from accessing it by using VLANs to split up the network.
- For security issues, VLANs provide granular control of the network because each VLAN is identified by a single IP subnetwork. All packets passing in and out of a VLAN are consistently tagged with the VLAN ID of that VLAN, thereby providing easy identification, because a VLAN ID on a packet cannot be altered. (For a switch that runs Junos OS that does not support ELS, we recommend that you avoid using 1 as a VLAN ID, because that ID is a default value.)
- VLANs react quickly to host relocation—this is also due to the persistent VLAN tag on packets.
- On an Ethernet LAN, all network nodes must be physically connected to the same network. In VLANs, the physical location of nodes is not important—you can group network devices in any way that makes sense for your organization, such as by department or business function, types of network nodes, or physical location.

Maximum VLANs and VLAN Members Per Switch

The number of VLANs supported per switch varies for each switch. Use the configuration-mode command **set vlans *vlan-name* vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because you have to assign a specific ID number when you create a VLAN—you could overwrite one of the numbers, but you cannot exceed the limit.

You can, however, exceed the recommended VLAN member maximum for a switch.

On a switch that runs Junos OS that does not support the ELS configuration style, the maximum number of VLAN members allowed on the switch is eight times the maximum number of VLANs that the switch supports (vmember limit = vlan max * 8). If the configuration of the switch exceeds the recommended VLAN member maximum, a

warning message appears when you commit the configuration. If you commit the configuration despite the warning, the commit succeeds, but there is a risk of the Ethernet switching process (eswd) failing as a result of memory allocation failure.

On a switch that runs Junos OS that supports ELS, the maximum number of VLAN members allowed on the switch is 24 times the maximum number of VLANs that the switch supports (vmember limit = vlan max * 24). If the configuration of the switch exceeds the recommended VLAN member maximum, a warning message appears in the system log (syslog).

A QFabric system supports up to 131,008 VLAN members (vmembers) on a single network node group, server node group, or redundant server node group. The number of vmembers is calculated by multiplying the maximum number of VLANs by 32.

For example, to calculate how many interfaces are required to support 4,000 VLANs, divide the maximum number of vmembers (128,000) by the number of configured VLANs (4,000). In this case, 32 interfaces are required.

On network Node groups and server Node groups, you can configure link aggregation groups (LAGs) across multiple interfaces. Each LAG and VLAN combination is considered a vmember.

A Default VLAN Is Configured on Most Switches

Some switches that run Junos OS that do not support the ELS configuration style are preconfigured with a VLAN named **default** that does not tag packets and operates only with untagged packets. On these switches, each interface already belongs to the VLAN named **default** and all traffic uses this VLAN until you configure more VLANs and assign traffic to those VLANs.

NOTE: When a Juniper Networks QFX3500 or QFX3600 switch is interconnected with other switches in a Virtual Chassis configuration, each individual switch that is included as a member of the configuration is identified with a member ID. The member ID functions as an FPC slot number. When you are configuring interfaces for a Virtual Chassis configuration, you specify the appropriate member ID (0 through 9) as the slot element of the interface name. The default factory settings for a Virtual Chassis configuration include FPC 0 as a member of the default VLAN because FPC 0 is configured as part of the ethernet-switching family. In order to include FPC 1 through FPC 9 in the default VLAN, add the ethernet-switching family to the configurations for those interfaces.

Assigning Traffic to VLANs

You can assign traffic on any switch to a particular VLAN by referencing either the interface port of the traffic or the MAC addresses of devices sending traffic.

Assign VLAN Traffic According to the Interface Port Source

This method is most commonly used to assign traffic to VLANs. In this case, you specify that all traffic received on a particular switch interface is assigned to a specific VLAN. You configure this VLAN assignment when you configure the switch, by using either the VLAN number (called a VLAN ID) or by using the VLAN name, which the switch then translates into a numeric VLAN ID. This method is referred to simply as creating a VLAN because it is the most commonly used method.

Assign VLAN Traffic According to the Source MAC Address

In this case, all traffic received from a specific MAC address is forwarded to a specific egress interface (next hop) on the switch. MAC-based VLANs are either static (named MAC addresses configured one at a time) or dynamic (configured using a RADIUS server).

To configure a static MAC-based VLAN on a switch that supports ELS, see *Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)*.To configure a static MAC-based VLAN on a switch that does not support ELS, see *Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)*.

Forwarding VLAN Traffic

To pass traffic within a VLAN, the switch uses Layer 2 forwarding protocols, including IEEE 802.1Q spanning-tree protocols.

To pass traffic between two VLANs, the switch uses standard Layer 3 routing protocols, such as static routing, OSPF, and RIP. The same interfaces that support Layer 2 bridging protocols also support Layer 3 routing protocols, providing multilayer switching.

To pass traffic from a single device on an access port to a switch and then pass those packets on a trunk port, use the native mode configuration previously discussed under "Trunk Mode" on page 8.

VLANs Communicate with Integrated Routing and Bridging Interfaces or Routed VLAN Interfaces

Traditionally, switches sent traffic to hosts that were part of the same broadcast domain (VLAN) but routers were needed to route traffic from one broadcast domain to another. Also, only routers performed other Layer 3 functions such as traffic engineering.

Switches that run Junos OS that supports the ELS configuration style perform inter-VLAN routing functions using an integrated routing and bridging (IRB) interface named irb, while switches that run Junos OS that does not support ELS perform these functions using a routed VLAN interface (RVI) named vlan. These interfaces detect both MAC addresses and IP addresses and route data to Layer 3 interfaces, thereby frequently eliminating the need to have both a switch and a router.

NOTE:

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102

- *Understanding FCoE*
- *Interfaces Overview*

Understanding Integrated Routing and Bridging

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs). VLANs limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN. For example, you might want to create a VLAN that includes the employees in a department and the resources that they use often, such as printers, servers, and so on.

Of course, you also want to allow these employees to communicate with people and resources in other VLANs. To forward packets between VLANs, you normally you need a router that connects the VLANs. However, you can accomplish this forwarding on a switch without using a router by configuring an integrated routing and bridging (IRB) interface. (These interfaces are also called routed VLAN interfaces, or RVIs). Using this approach reduces complexity and avoids the costs associated with purchasing, installing, managing, powering, and cooling another device.

An IRB is a special type of Layer 3 virtual interface named **vlan**. Like normal Layer 3 interfaces, the **vlan** interface needs a logical unit number with an IP address. In fact, to be useful an IRB needs at least two logical units and two IP addresses—you must create units with addresses in each of the subnets associated with the VLANs between which you want traffic to be routed. That is, if you have two VLANs (for example, VLAN **red** and VLAN **blue**) with corresponding subnets, your IRB must have a logical unit with an address in the subnet for **red** and a logical unit with an address in the subnet for **blue**. The switch automatically creates direct routes to these subnets and uses these routes to forward traffic between VLANs.

NOTE: If you are using a version of Junos OS that supports Enhanced Layer 2 Software (ELS), you can also create a Layer 3 virtual interface named **irb** instead of **vlan**—that is, both statements are supported by ELS

Table 3 on page 25 shows values you might use when configuring an IRB:

Table 3: Sample IRB Values

Property	Settings
VLAN names and tags (IDs)	**blue**, ID **100** **red**, ID **200**
Subnets associated with VLANs	**blue**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **red**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
IRB name	interface **irb**

Table 3: Sample IRB Values *(continued)*

Property	Settings
IRB units and addresses	logical unit 100: **192.0.2.1/25** logical unit 200: **192.0.2.129/25**

For the sake of consistency and to avoid confusion, Table 3 on page 25 shows IRB logical unit numbers that match the IDs of the corresponding VLANs. However, you do not have to assign logical unit numbers that match the VLAN IDs—you can use any values for the units. To bind the logical units of the IRB to the appropriate VLANs, you use the **l3-interface** statement.

Because IRBs operate at Layer 3, you can use Layer 3 services such as firewall filters or CoS rewriting with them.

Table 4 on page 26 shows the number of IRBs/RVIs that each QFX platform supports.

Table 4: Number of Supported IRBs/RVIs by Platform

Platform	Number of Supported IRBs/RVIs
QFX3500	1200
QFX3000-G	1024
QFX3000-M	1024

Related Documentation

- Example: Configuring Routing Between VLANs on One Switch on page 212

Understanding MAC Learning

MAC learning is the process of obtaining the MAC addresses of all the nodes on a network.

When a node is first connected to an Ethernet LAN or VLAN, it has no information about the other nodes on the network. As data is sent through the network, data packets include a data frame listing their source and destination MAC addresses. The data frame is forwarded to a target port, which is connected to the second device. The MAC address is learned locally at the target port, which facilitates communications for frames that later enter the target port and contain addresses previously learned from a received frame.

MAC learning can also be enabled on a per-VLAN basis. See"Example: Disabling MAC Learning in a VLAN" on page 110 for further information.

By default, MAC learning is enabled on the QFX Series.

Related Documentation

- Introduction to the Media Access Control (MAC) Layer 2 Sublayer on page 13
- Overview of Layer 2 Networking on page 3

Understanding Reflective Relay for Use with VEPA Technology

Virtual Ethernet Port Aggregator (VEPA) technology aggregates packets generated by virtual machines located on the same server and relays them to a physical switch. The physical switch then provides connectivity between the virtual machines located on the server, so the virtual machines do not communicate with one another. Offloading switching activities from a virtual switch to a physical switch reduces the computing overhead on the virtual servers and takes advantage of the security, filtering, and management features of the physical switch. Reflective relay, also known as "hairpin turn," enables the physical switch to receive aggregated packets from the virtual machines hosted on the server through the VEPA on the downstream port and send those packets out the same downstream port from which the physical switch received them.

- VEPA on page 27
- Reflective Relay on page 27

VEPA

Even though virtual machines are capable of sending packets directly to one another, it is more efficient to pass these aggregated packets from the VEPA to a physical switch. The switch can then send any packets destined for a virtual machine located on the same server to the VEPA.

Reflective Relay

Reflective relay, also known as a "hairpin turn" or "hairpin mode," returns aggregated packets to the VEPA by using the same downstream port that initially delivered the aggregated packets from the VEPA to the switch. Reflective relay must be configured on the interface located on the physical switch that receives aggregated packets, such as VEPA packets, because some of these packets might need to be sent back to the server if they are destined for another virtual machine on the same server.

Reflective relay only occurs in two situations:

- When the destination address of the packet was learned on that downstream port
- When the destination has not yet been learned

Reflective relay does not otherwise change the operation of the switch. If the interface to which the virtual machine is connected and the MAC address of the virtual machine packet are not yet included in the Ethernet switching table for the virtual machine's associated VLAN, an entry is added. If the source MAC address of an incoming packet under the respective VLAN is not yet present in the Ethernet switching table, the switch floods the packet on all the other ports that are members of the same VLAN, including the port on which the packet arrived.

Related Documentation

- *Understanding Bridging*
- Understanding Bridging and VLANs on page 5
- Example: Configuring Reflective Relay for Use with VEPA Technology on page 155

Understanding Private VLANs

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by splitting the broadcast domain into multiple isolated broadcast subdomains and essentially putting secondary VLANs inside a primary VLAN. PVLANs restrict traffic flows through their member switch ports (called “private ports”) so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. The uplink trunk port is usually connected to a router, firewall, server, or provider network. Each PVLAN typically contains many private ports that communicate only with a single uplink, thereby preventing the ports from communicating with each other.

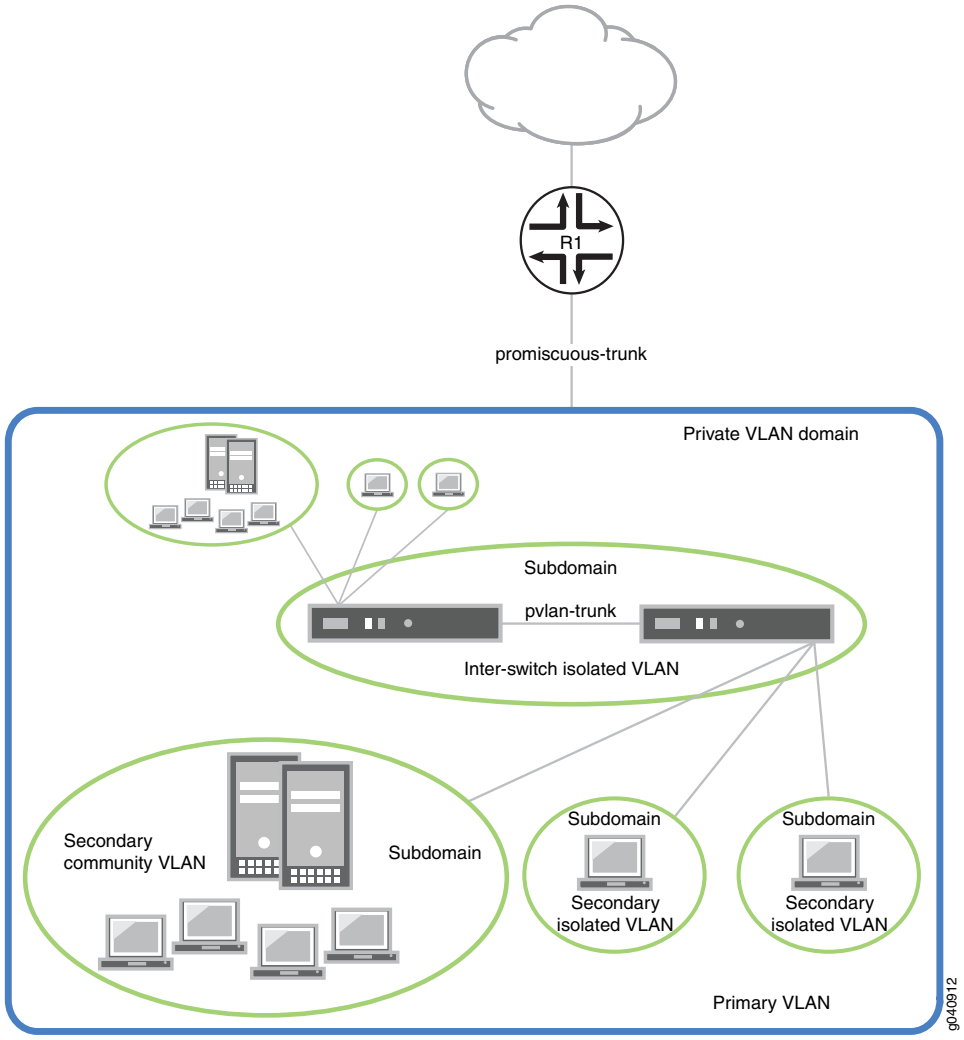
Just like regular VLANs, PVLANs are isolated on Layer 2 and require that a Layer 3 device be used to route traffic among them. PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts. Service providers use PVLANs to keep their customers isolated from one another.

This topic explains the following concepts regarding PVLANs on the QFX Series:

Typical Structure and Primary Application of PVLANs

A PVLAN can be created on a single switch or can be configured to span multiple switches. The PVLAN shown in Figure 1 on page 29 includes two switches, with a primary PVLAN domain and various subdomains.

Figure 1: Subdomains in a PVLAN



As shown in Figure 1 on page 29, a PVLAN has only one primary domain and multiple secondary domains. The types of domains are:

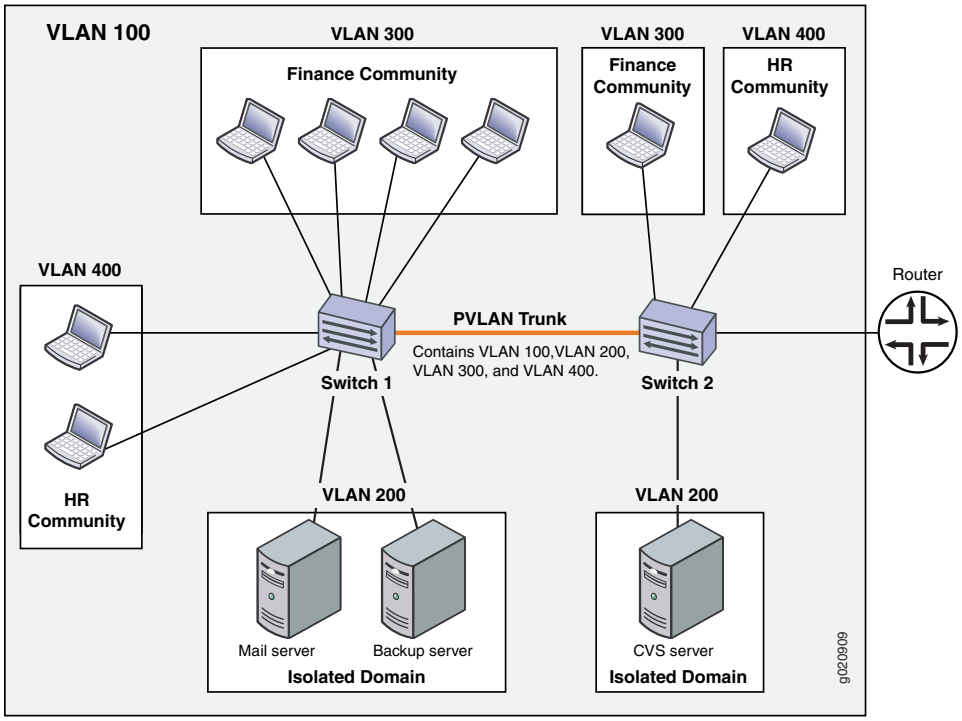
- Primary VLAN—VLAN used to forward frames downstream to isolated and community VLANs.
- Secondary isolated VLAN—VLAN that receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN.
- Secondary interswitch isolated VLAN—VLAN used to forward isolated VLAN traffic from one switch to another through PVLAN trunk ports. 802.1Q tags are required for interswitch isolated VLANs because IEEE 802.1Q uses an internal tagging mechanism

by which a trunking device inserts a 4-byte VLAN frame identification tab into the packet header.

- Secondary community VLAN—VLAN used to transport frames among members of a community (a subset of users within the VLAN) and to forward frames upstream to the primary VLAN.

Figure 2 on page 30 shows a PVLAN spanning multiple switches, where the primary VLAN (**100**) contains two community domains (**300** and **400**) and one interswitch isolated domain.

Figure 2: PVLAN Spanning Multiple Switches



NOTE: Primary and secondary VLANs count against the limit of 4089 VLANs supported on the QFX Series. For example, each VLAN in Figure 2 on page 30 counts against this limit.

Using 802.1Q Tags to Identify Packets

When packets are marked with a customer-specific 802.1Q tag, that tag identifies ownership of the packets for any switch or router in the network. Sometimes, 802.1Q tags are needed within PVLANs to keep track of packets from different subdomains. Table 5 on page 31 indicates when a VLAN 802.1Q tag is needed on the primary VLAN or on secondary VLANs.

Table 5: PVLAN Requirements for 802.1Q Tags

	On a Single Switch	On Multiple Switches
Primary VLAN	Specify an 802.1Q tag by setting a VLAN ID.	Specify an 802.1Q tag by setting a VLAN ID.
Secondary VLAN	No tag needed on VLANs.	VLANs need 802.1Q tags: • Specify an 802.1Q tag for each community VLAN by setting a VLAN ID. • Specify the 802.1Q tag for an isolation VLAN ID by setting an isolation ID.

Efficient Use of IP Addresses

PVLANs provide IP address conservation and efficient allocation of IP addresses. In a typical network, VLANs usually correspond to a single IP subnet. In PVLANs, the hosts in all secondary VLANs belong to the same IP subnet because the subnet is allocated to the primary VLAN. Hosts within the secondary VLAN are assigned IP addresses based on IP subnets associated with the primary VLAN, and their IP subnet masking information reflects that of the primary VLAN subnet. However, each secondary VLAN is a separate broadcast domain.

PVLAN Port Types

PVLANs can use six different port types. The network depicted in Figure 2 on page 30 uses a promiscuous port to transport information to the router, community ports to connect the finance and HR communities to their respective switches, isolated ports to connect the servers, and a PVLAN trunk port to connect the two switches. PVLAN ports have different restrictions:

- Promiscuous trunk port—A promiscuous port is an upstream trunk port connected to a router, firewall, server, or provider network. A promiscuous trunk port can communicate with all interfaces, including the isolated and community ports within a PVLAN.
- PVLAN trunk port—A PVLAN trunk port is required in multiswitch PVLAN configurations to span the switches. The PVLAN trunk port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the interswitch isolated VLAN), and it carries traffic from the primary VLAN and all secondary VLANs. It can communicate with all ports.

  Communication between a PVLAN trunk port and an isolated port is usually unidirectional. A PVLAN trunk port's membership in the interswitch isolated VLAN is egress-only, meaning that an isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port does not forward packets to an isolated port (unless the packets ingressed on a promiscuous access port and are therefore being forwarded to all the secondary VLANs in the same primary VLAN as the promiscuous port).
- Secondary VLAN trunk port (not shown)—Secondary trunk ports carry secondary VLAN traffic. For a given private VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different

primary VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

- Community port—Community ports communicate among themselves and with their promiscuous ports. Community ports serve only a select group of users. These interfaces are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.
- Isolated access port—Isolated ports have Layer 2 connectivity only with promiscuous ports and PVLAN trunk ports—an isolated port cannot communicate with another isolated port even if these two ports are members of the same isolated VLAN (or interswitch isolated VLAN) domain. Typically, a server, such as a mail server or a backup server, is connected on an isolated port. In a hotel, each room would typically be connected on an isolated port, meaning that room-to-room communication is not possible, but each room can access the Internet on the promiscuous port.
- Promiscuous access port (not shown)—These ports carry untagged traffic. Traffic that ingresses on a promiscuous access port is forwarded to all secondary VLAN ports on the device. If traffic ingresses into the device on a VLAN-enabled port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.

Table 6 on page 32 summarizes whether Layer 2 connectivity exists between the different types of ports.

Table 6: PVLAN Ports and Layer 2 Connectivity

Port Type	Promiscuous Trunk	PVLAN Trunk	Secondary Trunk	Community	Isolated Access	Promiscuous access
Promiscuous trunk	Yes	Yes	Yes	Yes	Yes	Yes
PVLAN trunk	Yes	Yes	Yes	Yes—same community only	Yes	Yes
Secondary Trunk	Yes	Yes	No	Yes	No	Yes
Community	Yes	Yes	Yes	Yes—same community only	No	Yes
Isolated access	Yes	Yes—unidirectional only	No	No	No	Yes
Promiscuous access	Yes	Yes	Yes	Yes	Yes	No

NOTE: If you enable the **no-mac-learning** statement on a primary VLAN, all isolated VLANs in the PVLAN inherit that setting. However, if you want to disable MAC address learning on any community VLANs, you must configure the **no-mac-learning** statement on each of those VLANs.

Limitations of Private VLANs

The following constraints apply to private VLAN configurations:

- IGMP snooping is not supported with private VLANs.
- Routed VLAN interfaces are not supported on private VLANs
- Routing between secondary VLANs in the same primary VLAN is not supported.

Related Documentation

- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 37
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

Understanding PVLAN Traffic Flows Across Multiple Switches

This topic illustrates and explains three different traffic flows on a sample multiswitch network configured with a private VLAN (PVLAN). PVLANs restrict traffic flows through their member switch ports (which are called “private ports”) so that they communicate only with a specific uplink trunk port or with specified ports within the same VLAN.

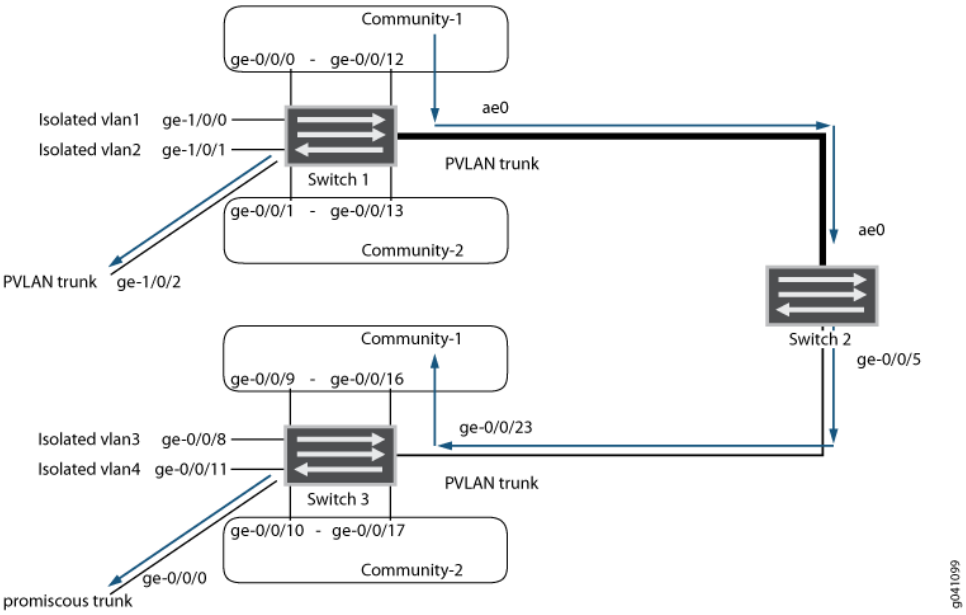
This topic describes:

- Community VLAN Sending Untagged Traffic on page 33
- Isolated VLAN Sending Untagged Traffic on page 34
- PVLAN Tagged Traffic Sent on a Promiscuous Port on page 35

Community VLAN Sending Untagged Traffic

In this scenario, a VLAN in Community-1 of Switch 1 at interface ge-0/0/0 sends untagged traffic. The arrows in Figure 3 on page 34 represent this traffic flow.

Figure 3: Community VLAN Sends Untagged Traffic



In this scenario, the following activity takes place on Switch 1:

- Community-1 VLAN on interface ge-0/0/0: Learning
- pvlan100 on interface ge-0/0/0: Replication
- Community-1 VLAN on interface ge-0/0/12: Receives traffic
- PVLAN trunk port: Traffic exits from ge-1/0/2 and from ae0 with tag 10
- Community-2: Interface receives no traffic
- Isolated VLANs: Interfaces receive no traffic

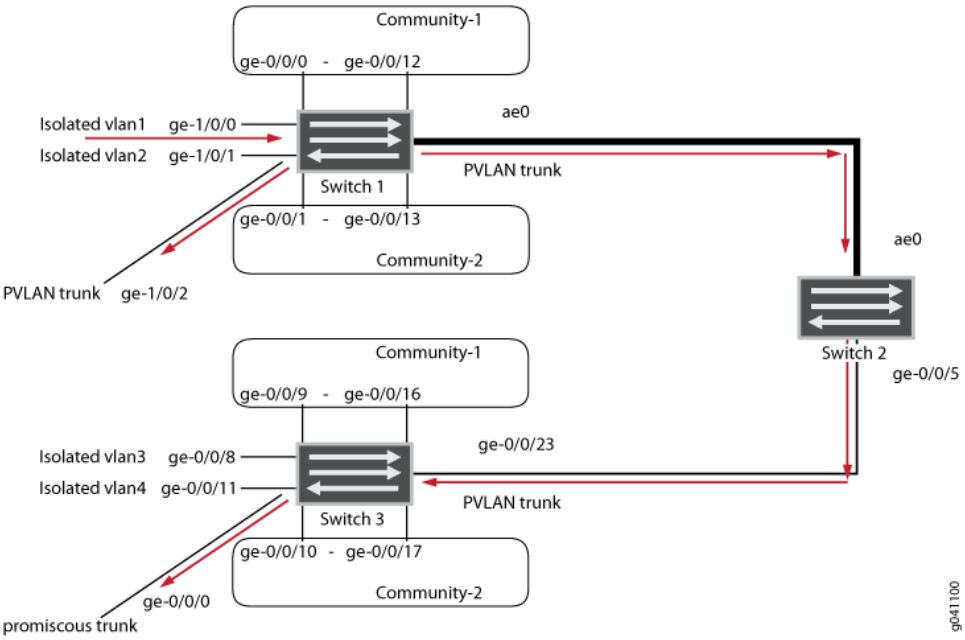
In this scenario, this activity takes place on Switch 3:

- Community-1 VLAN on interface ge-0/0/23 (PVLAN trunk): Learning
- pvlan100 on interface ge-0/0/23: Replication
- Community-1 VLAN on interface ge-0/0/9 and ge-0/0/16: Receives traffic
- Promiscuous trunk port: Traffic exits from ge-0/0/0 with tag 100
- Community-2: Interface receives no traffic
- Isolated VLANs: Interfaces receive no traffic

Isolated VLAN Sending Untagged Traffic

In this scenario, isolated VLAN1 on Switch 1 at interface ge-1/0/0 sends untagged traffic. The arrows in Figure 4 on page 35 represent this traffic flow.

Figure 4: Isolated VLAN Sends Untagged Traffic



In this scenario, the following activity takes place on Switch 1:

- Isolated VLAN1 on interface ge-1/0/0: Learning
- pvlan100 on interface ge-1/0/0: Replication
- Traffic exits from pvlan-trunk ge-1/0/2 and ae0 with tag 50
- Community-1 and Community-2: Interfaces receive no traffic
- Isolated VLANs: Interfaces receive no traffic

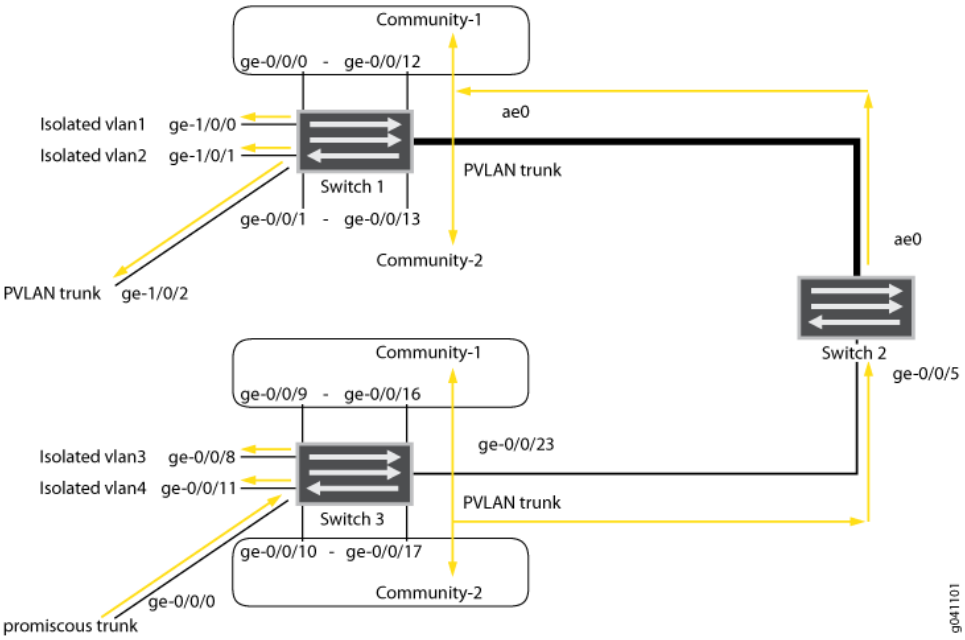
In this scenario, this activity takes place on Switch 3:

- VLAN on interface ge-0/0/23 (PVLAN trunk port): Learning
- pvlan100 on interface ge0/0/23: Replication
- Promiscuous trunk port: Traffic exits from ge-0/0/0 with tag 100
- Community-1 and Community-2: Interfaces receive no traffic
- Isolated VLANs: Receive no traffic

PVLAN Tagged Traffic Sent on a Promiscuous Port

In this scenario, PVLAN tagged traffic is sent on a promiscuous port. The arrows in Figure 5 on page 36 represent this traffic flow.

Figure 5: PVLAN Tagged Traffic Sent on a Promiscuous Port



In this scenario, the following activity takes place on Switch 1:

- pvlan100 VLAN on interface ae0 (PVLAN trunk): Learning
- Community-1, Community-2, and all isolated VLANs on interface ae0: Replication
- VLAN on interface ae0: Replication
- Traffic exits from pvlan-trunk ge-1/0/2 with tag 100
- Community-1 and Community-2: Interfaces receive traffic
- Isolated VLANs: Receive traffic

In this scenario, this activity takes place on Switch 3:

- pvlan100 on interface ge-0/0/0: Learning
- Community-1, Community-2 and all isolated VLANs on interface ge-0/0/0: Replication
- VLAN on interface ge-0/0/0: Replication
- Community-1 and Community-2: Interfaces receive traffic
- Isolated VLANs: Receive traffic

Related Documentation

- *Understanding Private VLANs on EX Series Switches*
- *Example: Configuring a Private VLAN on a Single EX Series Switch*
- *Example: Configuring a Private VLAN Spanning Multiple EX Series Switches*
- *Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)*
- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*

- Understanding Private VLANs on page 28
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249
- Example: Configuring a Private VLAN on a Single Switch on page 119
- Example: Configuring a Private VLAN Spanning Multiple Switches on page 124

Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by splitting a VLAN into multiple broadcast subdomains and essentially putting secondary VLANs inside a primary VLAN. PVLANs restrict traffic flows through their member ports so that these ports communicate only with a specified uplink trunk port or with specified ports within the same VLAN. The uplink trunk port is usually connected to a router, firewall, server, or provider network. A PVLAN typically contains many private ports that communicate only with a single uplink, thereby preventing the ports from communicating with each other.

Secondary trunk ports and promiscuous access ports extend the functionality of PVLANs for use in complex deployments, such as:

- Enterprise VMWare Infrastructure environments
- Multitenant cloud services with VM management
- Web hosting services for multiple customers

For example, you can use secondary VLAN trunk ports to connect QFX devices to VMware servers that are configured with private VLANs. You can use promiscuous access ports to connect QFX devices to systems that do not support trunk ports but do need to participate in private VLANs.

This topic explains the following concepts regarding PVLANs on the QFX Series:

- PVLAN Port Types on page 37
- Secondary VLAN Trunk Port Details on page 38
- Use Cases on page 39

PVLAN Port Types

PVLANs can use the following different port types:

- Promiscuous trunk port—A promiscuous port is an upstream trunk port connected to a router, firewall, server, or provider network. A promiscuous trunk port can communicate with all interfaces, including the isolated and community ports within a PVLAN.
- PVLAN trunk port—A PVLAN trunk port is required in multiswitch PVLAN configurations to span the switches. The PVLAN trunk port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the interswitch isolated VLAN),

and it carries traffic from the primary VLAN and all secondary VLANs. It can communicate with all ports.

Communication between a PVLAN trunk port and an isolated port is usually unidirectional. A PVLAN trunk port's membership in the interswitch isolated VLAN is egress-only, meaning that an isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port does not forward packets to an isolated port (unless the packets ingressed on a promiscuous access port and are therefore being forwarded to all the secondary VLANs in the same primary VLAN as the promiscuous port).

- Secondary VLAN trunk port—Secondary VLAN trunk ports carry secondary VLAN traffic. For a given private (primary) VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different primary VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

  NOTE: When traffic egresses from a secondary VLAN trunk port, it normally carries the tag of the primary VLAN that the secondary port is a member of. If you want traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag, use the extend-secondary-vlan-id statement.

- Community port—Community ports communicate among themselves and with their promiscuous ports. Community ports serve only a select group of users. These interfaces are separated at Layer 2 from all other interfaces in other communities or isolated ports within their PVLAN.
- Isolated access port—Isolated ports have Layer 2 connectivity only with promiscuous ports and PVLAN trunk ports. An isolated access port cannot communicate with another isolated port even if these two ports are members of the same isolated VLAN.
- Promiscuous access port—These ports carry untagged traffic and can be a member of only one primary VLAN. Traffic that ingresses on a promiscuous access port is forwarded to the ports of the secondary VLANs that are members of the primary VLAN that the promiscuous access port is a member of. In this case, the traffic carries the appropriate secondary VLAN tag when it egresses from the secondary VLAN port if the secondary VLAN port is a trunk port. If traffic ingresses on a secondary VLAN port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.

Secondary VLAN Trunk Port Details

When using a secondary VLAN trunk port, be aware of the following:

- You must configure an isolation VLAN ID for each primary VLAN that the secondary VLAN trunk port will participate in. This is true even if the secondary VLANs that the secondary VLAN trunk port will carry are confined to a single device.
- If you configure a port to be a secondary VLAN trunk port for a given primary VLAN, you can also configure the same physical port to be any of the following:

  - Secondary VLAN trunk port for another primary VLAN
  - PVLAN trunk for another primary VLAN
  - Promiscuous trunk port
  - Access port for a non-private VLAN
- Traffic that ingresses on a secondary VLAN trunk port (with a secondary VLAN tag) and egresses on a PVLAN trunk port retains the secondary VLAN tag on egress.
- Traffic that ingresses on a secondary VLAN trunk port and egresses on a promiscuous trunk port has the appropriate primary VLAN tag on egress.
- Traffic that ingresses on a secondary VLAN trunk port and egresses on a promiscuous access port is untagged on egress.
- Traffic that ingresses on a promiscuous trunk port with a primary VLAN tag and egresses on a secondary VLAN trunk port carries the appropriate secondary VLAN tag on egress. For example, assume that you have configured the following on a switch:
  - Primary VLAN 100
  - Community VLAN 200 as part of the primary VLAN
  - Promiscuous trunk port
  - Secondary trunk port that carries community VLAN 200

  If a packet ingresses on the promiscuous trunk port with primary VLAN tag 100 and egresses on the secondary VLAN trunk port, it carries tag 200 on egress.

Use Cases

On the same physical interface, you can configure multiple secondary VLAN trunk ports (in different primary VLANs) or combine a secondary VLAN trunk port with other types of VLAN ports. The following use cases provide examples of doing this and show how traffic would flow in each case:

- Secondary VLAN Trunks In Two Primary VLANS on page 39
- Secondary VLAN Trunk and Promiscuous Trunk on page 41
- Secondary VLAN Trunk and PVLAN Trunk on page 42
- Secondary VLAN Trunk and Non-Private VLAN Interface on page 44
- Traffic Ingressing on Promiscuous Access Port on page 45

Secondary VLAN Trunks In Two Primary VLANS

For this use case, assume you have two switches with the following configuration:

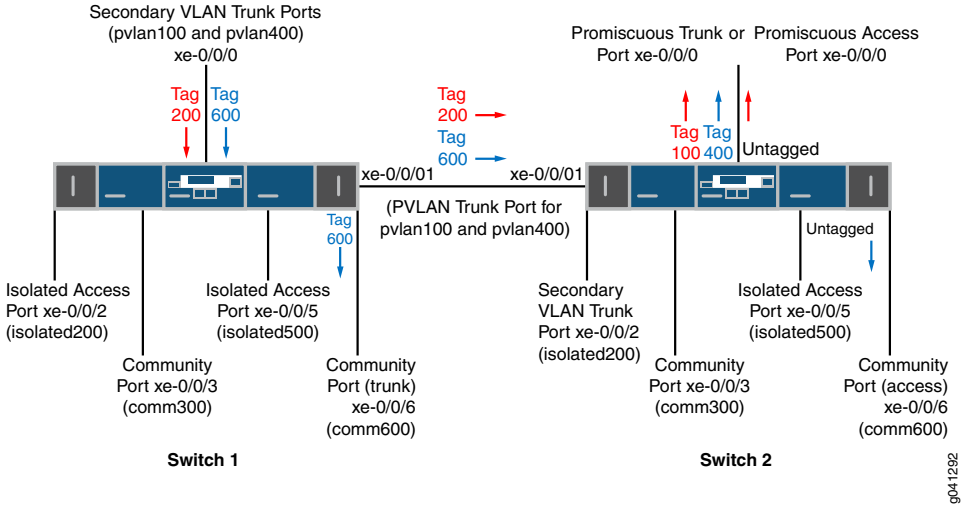
- Primary VLAN pvlan100 with tag 100.
  - Isolated VLAN isolated200 with tag 200 is a member of pvlan100.
  - Community VLAN comm300 with tag 300 is a member of pvlan100.
- Primary VLAN pvlan400 with tag 400.

- Isolated VLAN isolated500 with tag 500 is a member of pvlan400.
- Community VLAN comm600 with tag 600 is a member of pvlan400.

- Interface xe-0/0/0 on Switch 1 connects to a VMware server (not shown) that is configured with the private VLANs used in this example. This interface is configured with secondary VLAN trunk ports to carry traffic for secondary VLAN comm600 and the isolated VLAN (tag 200) that is a member of pvlan100.
- Interface xe-0/0/0 on Switch 2 is shown configured as a promiscuous trunk port or promiscuous access port. In the latter case, you can assume that it connects to a system (not shown) that does not support trunk ports but is configured with the private VLANs used in this example.
- On Switch 1, xe-0/0/6 is a member of comm600 and is configured as a trunk port.
- On Switch 2, xe-0/0/6 is a member of comm600 and is configured as an access port.

Figure 6 on page 40 shows this topology and how traffic for isolated200 and comm600 would flow after ingressing on xe-0/0/0 on Switch 1. Note that traffic would flow only where the arrows indicate. For example, there are no arrows for interfaces xe-0/0/2, xe-0/0/3, and xe-0/0/5 on Switch 1 because no packets would egress on those interfaces.

Figure 6: Two Secondary VLAN Trunk Ports on One Interface



Here is the traffic flow for VLAN isolated200:

1. After traffic for isolated200 ingresses on the secondary VLAN trunk port on Switch 1, it egresses on the PVLAN trunk port because the PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (200) when egressing.
2. After traffic for isolated200 ingresses on the secondary VLAN trunk port on Switch 2, it egresses on xe-0/0/0, which is configured as a promiscuous trunk port or promiscuous access port.
   - If xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port, the packets egress on this port with the primary VLAN tag (100).

- If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the packets egress on this port untagged.

Note that traffic for VLAN isolated200 does not egress on isolated access port xe-0/0/2 on Switch 1 or secondary VLAN trunk port xe-0/0/2 on Switch 2 even though these two ports are members of the same isolated VLAN.

Here is the traffic flow for VLAN comm600:

1. After traffic for comm600 ingresses on the secondary VLAN trunk port on Switch 1, it egresses on the PVLAN trunk port because the PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (600) when egressing.
2. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 1. The traffic is tagged because the port is configured as a trunk.
3. After traffic for comm600 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, if this interface is configured as a promiscuous trunk port.

   NOTE: If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the port can participate in only one primary VLAN. In this case, the promiscuous access port is part of pvlan100, so traffic for comm600 does not egress from it

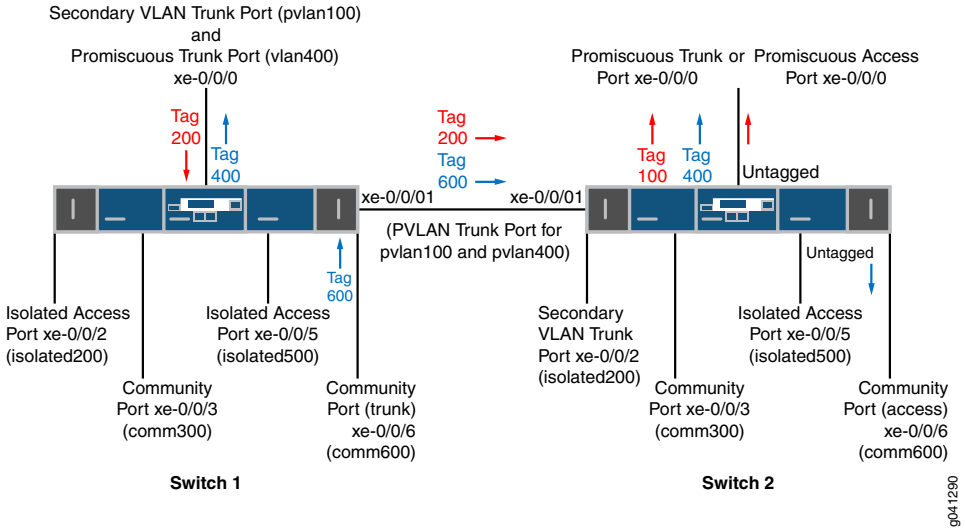
4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2. In this case, the traffic is untagged because the port mode is access.

Secondary VLAN Trunk and Promiscuous Trunk

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use case, with one exception: In this case, xe-0/0/0 on Switch 1 is configured as a secondary VLAN trunk port for VLAN pvlan100 and is also configured as a promiscuous trunk port for pvlan400.

Figure 7 on page 42 shows this topology and how traffic for isolated200 (member of pvlan100) and comm600 (member of pvlan400) would flow after ingressing on Switch 1.

Figure 7: Secondary VLAN Trunk and Promiscuous Trunk on One Interface



The traffic flow for VLAN isolated200 is the same as in the previous use case, but the flow for comm600 is different. Here is the traffic flow for VLAN comm600:

1. After traffic for comm600 ingresses on community VLAN port xe-0/0/6 on Switch 1, it egresses on promiscuous trunk port xe-0/0/0 on Switch 1. In this case it carries the primary VLAN tag (400).
2. Traffic for comm600 also egresses on the PVLAN trunk port because the PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (600) when egressing.
3. After traffic for comm600 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, if this interface is configured as a promiscuous trunk port.

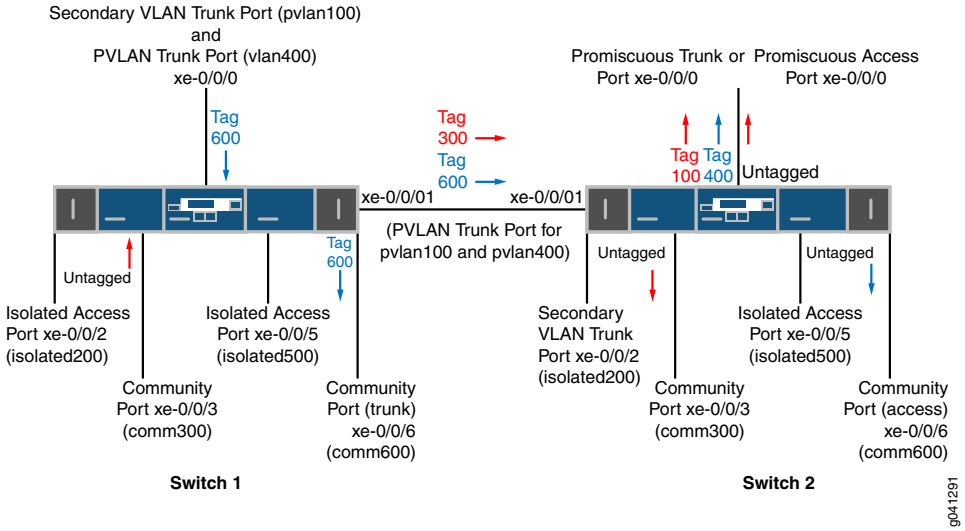
   It does not egress on xe-0/0/0 if this interface is configured as a promiscuous access port because the port can participate only in pvlan100.
4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2.

Secondary VLAN Trunk and PVLAN Trunk

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use cases except that xe-0/0/0 on Switch 1 is configured as a secondary VLAN trunk port for VLAN pvlan100 and is also configured as a PVLAN trunk port for pvlan400.

Figure 8 on page 43 shows this topology and how traffic for comm300 (member of pvlan100) and comm600 (member of pvlan400) would flow after ingressing on Switch 1.

Figure 8: Secondary VLAN Trunk and PVLAN Trunk on One Interface



Here is the traffic flow for VLAN comm300:

1. After traffic for comm300 ingresses on community port xe-0/0/3 on Switch 1, it egresses on PVLAN trunk port xe-0/0/1 because that PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (300) when egressing.

   NOTE: **Traffic for comm300 does not egress on xe-0/0/0 because the secondary VLAN trunk port on this interface carries isolated200, not comm300.**

2. After traffic for comm300 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, which is configured as a promiscuous trunk port or promiscuous access port.

   - If xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port, the packets egress on this port with the primary VLAN tag (100).
   - If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the packets egress on this port untagged.

3. Traffic for comm300 also egresses on community port xe-0/0/3 on Switch 2.

Here is the traffic flow for VLAN comm600:

1. After traffic for comm600 ingresses on the PVLAN port xe-0/0/0 on Switch 1, it egresses on the community port xe-0/0/6 on Switch 1. The packets keep the secondary VLAN tag (600) when egressing because xe-0/0/6 is a trunk port.
2. Traffic for comm600 also egresses on PVLAN trunk port xe-0/0/1 because that PVLAN trunk port is a member of all the VLANs. The packets keep the secondary VLAN tag (600) when egressing.

3. After traffic for comm600 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, if this interface is configured as a promiscuous trunk port.

   It does not egress on xe-0/0/0 if this interface is configured as a promiscuous access port because the port can participate only in pvlan100.

4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2. This traffic is untagged on egress because xe-0/0/6 is an access port.

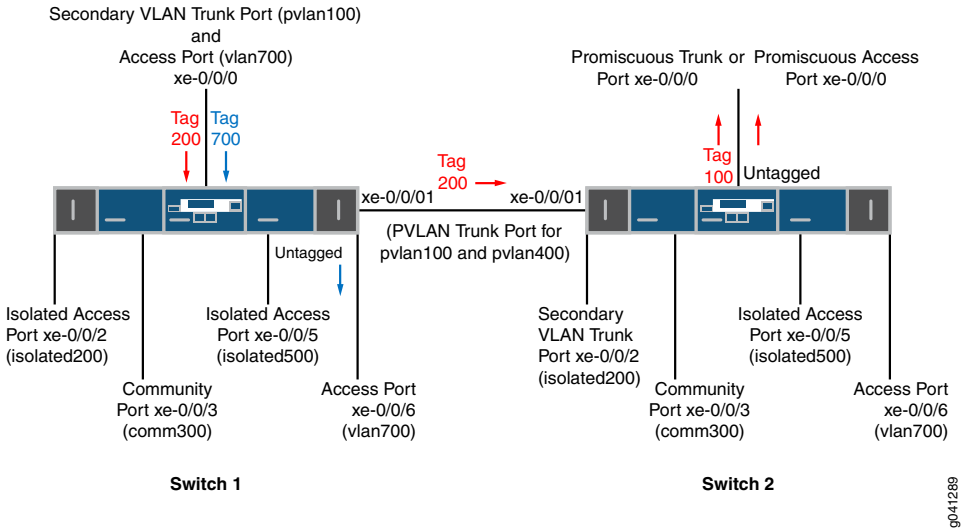
Secondary VLAN Trunk and Non-Private VLAN Interface

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use cases except for these differences:

- Configuration for xe-0/0/0 on Switch 1:
  - Secondary VLAN trunk port for VLAN pvlan100
  - Access port for vlan700
- Port xe-0/0/6 on both switches is an access port for vlan700.

Figure 9 on page 44 shows this topology and how traffic for isolated200 (member of pvlan100) and vlan700 would flow after ingressing on Switch 1.

Figure 9: Secondary VLAN Trunk and Non-Private VLAN Port on One Interface



Here is the traffic flow for VLAN isolated200:

1. After traffic for isolated200 ingresses on the secondary VLAN trunk port on Switch 1, it egresses on the PVLAN trunk port. The packets keep the secondary VLAN tag (200) when egressing.
2. After traffic for isolated200 ingresses on the PVLAN trunk port on Switch 2, it egresses on xe-0/0/0, which is configured as a promiscuous trunk port or promiscuous access port.

- If xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port, the packets egress on this port with the primary VLAN tag (100).
- If xe-0/0/0 on Switch 2 is configured as a promiscuous access port, the packets egress on this port untagged.

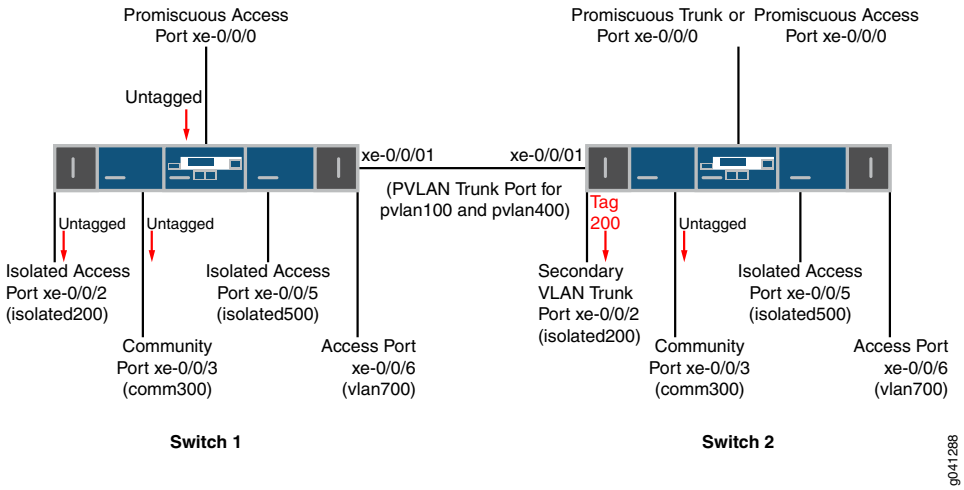
Note that traffic for VLAN isolated200 does not egress on isolated access port xe-0/0/2 on Switch 1 or secondary VLAN trunk port xe-0/0/2 on Switch 2 even though these two ports are members of the same isolated VLAN.

After traffic for vlan700 ingresses on the access port configured on xe-0/0/0 on Switch 1, it egresses on access port xe-0/0/6 because that port is a member of the same VLAN. Traffic for vlan700 is not forwarded to Switch 2 (even though xe-0/0/6 on Switch 2 is a member of vlan700) because the PVLAN trunk on xe-0/0/1 does not carry this VLAN.

Traffic Ingressing on Promiscuous Access Port

For this use case, assume you have two switches configured with the same ports and VLANs as in the previous use case except that xe-0/0/0 on Switch 1 is configured as a promiscuous access port and is a member of pvlan100. Figure 10 on page 45 shows this topology and how untagged traffic would flow after ingressing through this interface on Switch 1.

Figure 10: Traffic Ingressing on Promiscuous Access Port



As the figure shows, untagged traffic that ingresses on a promiscuous access port is forwarded to all the secondary VLAN ports that are members of the same primary VLAN that the promiscuous access port is a member of. The traffic is untagged when it egresses from access ports and tagged on egress from a trunk port (xe-0/0/2 on Switch 2).

Related Documentation

- Understanding Private VLANs on page 28
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on page 139
- Creating a Private VLAN on a Single Switch on page 247

- Creating a Private VLAN Spanning Multiple Switches on page 249
- Understanding Egress Firewall Filters with PVLANs on page 46
- *Troubleshooting Private VLANs*

Understanding Egress Firewall Filters with PVLANs

If you apply firewall filters to private VLANs in the output direction, the behavior of the filters might be unexpected. This topic explains how egress filters behave when applied to private VLANs.

If you apply a firewall filter in the output direction to a primary VLAN, the filter also applies to the secondary VLANs that are members of the primary VLAN when the traffic egresses with the primary VLAN tag or isolated VLAN tag, as listed below:

- Traffic forwarded from a secondary VLAN trunk port to a promiscuous port (trunk or access)
- Traffic forwarded from a secondary VLAN trunk port to a PVLAN trunk port.
- Traffic forwarded from a promiscuous port (trunk or access) to a secondary VLAN trunk port
- Traffic forwarded from a PVLAN trunk port. to a secondary VLAN trunk port
- Traffic forwarded from a community port to a promiscuous port (trunk or access)

If you apply a firewall filter in the output direction to a primary VLAN, the filter does *not* apply to traffic that egresses with a community VLAN tag, as listed below:

- Traffic forwarded from a community trunk port to a PVLAN trunk port
- Traffic forwarded from a promiscuous port (trunk or access) to a community trunk port
- Traffic forwarded from a PVLAN trunk port. to a community trunk port

If you apply a firewall filter in the output direction to a community VLAN, the following behaviors apply:

- The filter is applied to traffic forwarded from a promiscuous port (trunk or access) to a community trunk port (because the traffic egresses with the community VLAN tag).
- The filter is applied to traffic forwarded from a community port to a PVLAN trunk port (because the traffic egresses with the community VLAN tag).
- The filter is *not* applied to traffic forwarded from a community port to a promiscuous port (because the traffic egresses with the primary VLAN tag or untagged).

Related Documentation

- Understanding Private VLANs on page 28
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on page 139

- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249
- *Troubleshooting Private VLANs*

Understanding Multiple VLAN Registration Protocol (MVRP)

Multiple VLAN Registration Protocol (MVRP) is a Layer 2 messaging protocol that automates the creation and management of virtual LANs, thereby reducing the time you have to spend on these tasks. If your QFabric system connects to servers that host many virtual machines that require their own VLANs, using MVRP can save you the time and effort that would be required to manually create and administer the VLANs on the ports that connect to the servers. For example, if a virtual machine moves between servers—and therefore connects to a different redundant server Node group interface—MVRP can configure the appropriate VLAN membership on the new server Node group interface.

When using MVRP on a QFabric system, you must manually create on the QFabric the VLANs that exist on the attached servers because the QFabric implementation of MVRP does not allow VLANs to be created dynamically. However, you do not need to manually assign VLAN membership to the QFabric ports that connect to the servers. MVRP automatically assigns VLAN membership to server-facing QFabric ports when it learns about a VLAN from an attached server.

MVRP is an application protocol of the Multiple Registration Protocol (MRP) and is defined in the IEEE 802.1ak standard. MRP and MVRP replace Generic Attribute Registration Protocol (GARP) and GARP VLAN Registration Protocol (GVRP) and overcome GARP and GVRP limitations.

NOTE: **MVRP on QFabric systems does not support private VLANs.**

- QFabric Requirements on page 47
- MVRP Operations on page 48
- MRP Timers Control MVRP Updates on page 48
- MVRP Uses MRP Messages to Transmit Switch and VLAN States on page 49

QFabric Requirements

When configuring MVRP on a QFabric system, you can enable it globally or enable it only on the trunk ports that need to carry VLAN traffic from the attached servers. You also must manually create the expected VLANs, but you do not have to assign VLAN membership to the server-facing redundant server Node ports (as mentioned previously). However, you *do* have to manually assign VLAN membership to the uplink ports on the redundant server Node group and network Node group devices that will carry the VLAN traffic. Table 7 on page 48 summarizes the VLAN requirements for redundant server Node groups and network Node groups:

Table 7: MVRP VLAN Requirements for Node Devices

Node Group Type	Interface	Assign VLAN Membership to Trunk Ports?
Redundant server Node group	Server-facing trunk	No
Redundant server Node group	Uplink trunk (to interconnect device)	Yes
Network Node groups	Uplink trunk (to interconnect device)	Yes

MVRP Operations

MVRP stays synchronized by using MVRP protocol data units (PDUs). These PDUs specify which QFabric systems and switches are members of which VLANs, and which switch interfaces are in each VLAN. The MVRP PDUs are sent to other switches in the QFabric system when an MVRP state change occurs, and the receiving switches update their MVRP states accordingly. MVRP timers dictate when PDUs can be sent and when switches receiving MVRP PDUs can update their MVRP information.

In addition to this behavior, QFX switches include a mode—called passive mode—in which an MVRP-configured interface does not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from a peer (server) on that interface. By default MVRP-configured interfaces behave in the standard manner and automatically send PDU updates to announce any VLAN changes. (This is called active mode.)

To enable passive mode on an interface, enter and commit this statement:

set protocols mvrp interface *interface-name* passive

To keep VLAN membership information current, MVRP removes switches and interfaces when they become unavailable. Pruning VLAN information has these benefits:

- Limits the network VLAN configuration to active participants, thereby reducing network overhead.
- Limits broadcast, unknown unicast, and multicast (BUM) traffic to interested devices.

MVRP is disabled by default and is valid only for trunk interfaces.

MRP Timers Control MVRP Updates

MVRP registration and updates are controlled by timers that are part of MRP. The timers define when MVRP PDUs can be sent and when MVRP information can be updated. You configure the timers on a per-interface basis.

The following MRP timers are used to control the operation of MVRP:

- Join timer—Controls the interval for the next MVRP PDU transmit opportunity.
- Leave timer—Controls the period of time that an interface on the switch waits in the leave state before changing to the unregistered state.

- LeaveAll timer—Controls the frequency with which the interface generates LeaveAll messages.

BEST PRACTICE: **Unless there is a compelling reason to change the timer settings, leave the default settings in place. Modifying timers to inappropriate values can cause an imbalance in the operation of MVRP.**

MVRP Uses MRP Messages to Transmit Switch and VLAN States

MVRP uses MRP messages to register and declare MVRP states for a interface or VLAN and to inform the switching network that a interface or VLAN is leaving MVRP. These messages are communicated in the MRP PDUs sent by MVRP-enabled interfaces.

The following MRP messages are communicated for MVRP:

- Empty—MVRP information is not declared and no VLAN is registered.
- In—MVRP information is not declared but a VLAN is registered.
- JoinEmpty—MVRP information is declared but no VLAN is registered.
- JoinIn—MVRP information is declared and a VLAN is registered.
- Leave—MVRP information that was previously declared is withdrawn.
- LeaveAll—Unregister all VLANs on the switch. VLANs must re-register to participate in MVRP.
- New—The MVRP information is new and a VLAN might not be registered yet.

Related Documentation

- Understanding Bridging and VLANs on page 5
- Example: Configuring Automatic VLAN Administration Using MVRP on page 71
- Configuring Multiple VLAN Registration Protocol on page 243

CHAPTER 3

Spanning Trees Overview

Overview of Spanning-Tree Protocols

QFX Series switches provide Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), Multiple Spanning Tree Protocol (MSTP), and VLAN Spanning Tree Protocol (VSTP). The default spanning-tree protocol on the QFX Series is RSTP. RSTP provides faster convergence times than STP. However, some legacy networks require the slower convergence times of basic STP.

The STP support provided for the QFX Series includes:

- IEEE 802.1d
- 802.1w RSTP
- 802.1s MSTP

If your network includes IEEE 802.1D 1998 bridges, you can remove RSTP and explicitly configure STP. See "Configuring STP" on page 259. When you explicitly configure STP, the QFX Series products use the IEEE 802.1D 2004 specification, force version 0. This configuration runs a version of RSTP that is compatible with the classic, basic STP. If you use virtual LANs (VLANs), you should enable VSTP and use it on your network. See "Understanding VSTP" on page 54.

You can use the same operational commands (**show spanning-tree bridge** and **show spanning-tree interface**) to check the status of your spanning-tree configuration, regardless of which spanning-tree protocol has been configured.

STP uses bridge protocol data unit (BPDU) packets to exchange information with other switches. BPDUs send hello packets out at regular intervals to exchange information across bridges and detect loops in a network topology. There are two types of BPDUs:

- Configuration BPDUs—These BPDUs contain configuration information about the transmitting switch and its ports, including switch and port MAC addresses, switch priority, port priority, and port cost.
- Topology change notification (TCN) BPDUs—When a bridge needs to signal a topology change, it starts to send TCNs on its root port. The designated bridge receives the TCN, acknowledges it, and generates another one for its own root port. The process continues until the TCN reaches the root bridge.

STP uses the information provided by the BPDUs to elect a root bridge, identify root ports for each switch, identify designated ports for each physical LAN segment, and prune specific redundant links to create a loop-free tree topology. All leaf devices calculate the best path to the root device and place their ports in blocking or forwarding states based on the best path to the root. The resulting tree topology provides a single active Layer 2 data path between any two end stations.

Understanding Spanning Tree Protocols on a QFabric System

Although there is no need to run STP in a QFabric system, you can connect a QFabric system to another Layer 2 device and use STP. STP traffic can only be processed on network Node groups. Other Node groups, such redundant server Node groups and server Node groups, discard the STP bridge protocol data units (BPDUs) traffic and disable the interface automatically. Server Node groups only process host-facing protocols, whereas Network Node groups process all supported protocols.

Related Documentation

- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55
- Understanding MSTP on page 52
- Understanding RSTP on page 53
- Understanding VSTP on page 54

Understanding MSTP

Although RSTP provides faster convergence time than STP does, it still does not solve a problem inherent in STP: all VLANs within a LAN must share the same spanning tree. To solve this problem, the QFX Series products use Multiple Spanning Tree Protocol (MSTP) to create a loop-free topology in networks with multiple spanning-tree regions.

An MSTP region allows a group of bridges to be modeled as a single bridge. An MSTP region contains multiple spanning-tree instances (MSTIs). MSTIs provide different paths for different VLANs. This functionality facilitates more efficient load sharing across redundant links.

An MSTP region can support up to 64 MSTIs, and each instance can support from 1 through 4094 VLANs.

Related Documentation

- Overview of Spanning-Tree Protocols on page 51
- Understanding RSTP on page 53
- Example: Configuring Network Regions for VLANs with MSTP on page 184

Understanding RSTP

Juniper Networks QFX Series products use Rapid Spanning Tree Protocol (RSTP) on the network side of the QFX Series to provide quicker convergence time than the base Spanning Tree Protocol (STP) does. RSTP identifies certain links as point to point. When a point-to-point link fails, the alternate link can transition to the forwarding state, which speeds up convergence.

Although STP provides basic loop prevention functionality, it does not provide fast network convergence when there are topology changes. The STP process to determine network state transitions is slower than the RSTP process because it is timer-based. A device must reinitialize every time a topology change occurs. The device must start in the listening state and transition to the learning state and eventually to a forwarding or blocking state. When default values are used for the maximum age (20 seconds) and forward delay (15 seconds), it takes 50 seconds for the device to converge. RSTP converges faster because it uses a handshake mechanism based on point-to-point links instead of the timer-based process used by STP.

For networks with virtual LANs (VLANs), you can use VLAN Spanning Tree Protocol (VSTP), which takes the paths of each VLAN into account when calculating routes. VSTP uses RSTP by default.

An RSTP domain running from the edge outward on a QFX Series product has the following components:

- A *root port*, which is the “best path” to the root device.
- A *designated port*, which indicates that the switch is the designated bridge for the other switch connecting to this port.
- An *alternate port*, which provides an alternate root port.
- A *backup port*, which provides an alternate designated port.

Port assignments change through messages exchanged throughout the domain. An RSTP device generates configuration messages once per hello time interval. If an RSTP device does not receive a configuration message from its neighbor after an interval of three hello times, it determines that the connection with the neighbor is lost. When a *root port* or a *designated port* fails on a device, the device generates a configuration message with the proposal bit set. Once its neighbor device receives this message, it verifies that this configuration message is valid for that port and starts a *synchronizing* operation to ensure that all of its ports are in sync with the new information.

Similar sets of messages propagate through the network, restoring the connectivity very quickly after a topology change (in a well-designed network that uses RSTP, network

convergence can take as little as 0.5 seconds). If a device does not receive an agreement to a proposal message it has sent, it returns to the original IEEE 802.D convention.

RSTP was originally defined in the IEEE 802.1w draft specification and later incorporated into the IEEE 802.1D-2004 specification.

VSTP and RSTP can be configured at the same time. If you configure VSTP and RSTP at the same time and the switch has more than 253 VLANs, VSTP is configured only for the first 253 VLANs. For the remaining VLANs, only RSTP is configured. RSTP and VSTP are the only spanning-tree protocols that can be configured at the same time on the QFX Series.

Related Documentation

- Overview of Spanning-Tree Protocols on page 51
- Understanding MSTP on page 52
- Understanding VSTP on page 54
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165

Understanding VSTP

VLAN Spanning Tree Protocol (VSTP) enables Juniper Networks switches to run one or more Spanning Tree Protocol (STP) or Rapid Spanning Tree Protocol (RSTP) instances for each VLAN on which VSTP is enabled. For networks with multiple VLANs, VSTP improves intelligent tree spanning by defining best paths within the VLANs instead of within the entire network.

You can configure VSTP for a maximum of 509 VLANs.

VSTP and RSTP can be configured at the same time. If you configure VSTP and RSTP at the same time and the switch has more than 253 VLANs, VSTP is configured only for the first 253 VLANs. For the remaining VLANs, only RSTP is configured. RSTP and VSTP are the only spanning-tree protocols that can be configured at the same time on a switch.

NOTE: **We recommend that you enable VSTP on all VLANs that could receive VSTP bridge protocol data units (BPDUs).**

Related Documentation

- Overview of Spanning-Tree Protocols on page 51
- Understanding RSTP on page 53
- Configuring VLAN Spanning Tree Protocol on page 261
- *Configuring VLAN Spanning-Tree Protocol*
- **vstp on page 348**

Understanding BPDU Protection for STP, RSTP, and MSTP

NOTE: Using the original CLI, you can disable BPDU protection on interfaces by issuing the **set ethernet-switching-options bpdu-block** *interface-name* **disable** command.

A Juniper Networks device Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), and Multiple Spanning Tree Protocol (MSTP). Bridge protocol data unit (BPDU) protection can help prevent STP misconfigurations that can lead to network outages.

A loop-free network is supported through the exchange of a special type of frame called a BPDU. Receipt of BPDUs on certain interfaces in an STP, RSTP, VSTP, or MSTP topology, however, can lead to network outages. Enable BPDU protection on those interfaces to prevent these outages.

Peer STP applications running on the device interfaces use BPDUs to communicate. Ultimately, the exchange of BPDUs determines which interfaces block traffic and which interfaces become root ports and forward traffic.

However, a user bridge application running on a device connected to the device can also generate BPDUs. If these BPDUs are picked up by STP applications running on the device, they can trigger STP miscalculations, and those miscalculations can lead to network outages.

Enable BPDU protection on device interfaces connected to user devices or on interfaces on which no BPDUs are expected, such as edge ports. If BPDUs are received on a protected interface, the interface is disabled and stops forwarding frames.

Not only can you configure BPDU protection on a device with a spanning tree, but also on a device without a spanning tree. This type of topology typically consists of a non-STP device connected to an STP device through a trunk interface.

To configure BPDU protection on a device with a spanning tree, include the **bpdu-block-on-edge** statement at the [**edit protocols** (**stp** | **mstp** | **rstp**)] hierarchy level. To configure BPDU protection on a device without a spanning tree, include the **bpdu-block** statement at the [**edit ethernet-switching-options interface** *interface-name*] hierarchy level.

If BPDUs are sent to an interface (indicating that the misconfiguration has been corrected), the interface can be unblocked in one of two ways:

- If the **disable-timeout** statement has been included in the BPDU configuration, the interface automatically returns to service after the timer expires.
- Use the operational mode command **clear ethernet-switching bpdu-error**.

Disabling the BPDU protection configuration does not unblock the interface.

Related Documentation

- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Understanding Loop Protection for STP, RSTP, VSTP, and MSTP on page 56
- Understanding Root Protection for STP, RSTP, VSTP, and MSTP on page 57
- Understanding MSTP on page 52
- Understanding RSTP on page 53
- Understanding VSTP on page 54

Understanding Loop Protection for STP, RSTP, VSTP, and MSTP

A Juniper Networks device provides Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), and Multiple Spanning Tree Protocol (MSTP). Loop protection increases the efficiency of STP, RSTP, and MSTP by preventing ports from entering a forwarding state that would cause a loop to open in the network.

A loop-free network in spanning-tree topologies is supported through the exchange of a special type of frame called a bridge protocol data unit (BPDU). Peer STP applications running on the device interfaces use BPDUs to communicate. Ultimately, the exchange of BPDUs determines which interfaces block traffic (preventing loops) and which interfaces become root ports and forward traffic.

However, a blocking interface can mistakenly transition to the forwarding state if the interface stops receiving BPDUs from its designated port on the segment. Such a transition error can occur when there is a hardware error on the device or software configuration error between the device and its neighbor.

When loop protection is enabled, the spanning-tree topology detects root ports and blocked ports and ensures that both keep receiving BPDUs. If a loop-protection-enabled interface stops receiving BPDUs from its designated port, it reacts as it would react to a problem with the physical connection on this interface. It does not transition the interface to a forwarding state, but instead transitions it to a loop-inconsistent state. The interface recovers and it transitions back to the spanning-tree blocking state as soon as it receives a BPDU.

We recommend that you enable loop protection on all device interfaces that have a chance of becoming root or designated ports. Loop protection is most effective when enabled in the entire switched network. When you enable loop protection, you must configure at least one action (**alarm**, **block**, or both).

An interface can be configured for either loop protection or root protection, but not for both.

Related Documentation

- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Understanding Root Protection for STP, RSTP, VSTP, and MSTP on page 57

- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55
- Understanding MSTP on page 52
- Understanding RSTP on page 53
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54

Understanding Root Protection for STP, RSTP, VSTP, and MSTP

A Juniper Networks device provides Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), and Multiple Spanning Tree Protocol (MSTP). A loop-free network is supported through the exchange of a special type of frame called a bridge protocol data unit (BPDU). Peer STP applications running on the device interfaces use BPDUs to communicate. Ultimately, the exchange of BPDUs determines which interfaces block traffic and which interfaces become root ports and forward traffic.

You can also see BPDUs generated when you run a bridge application on a device attached to the device. This can interfere with root port election, which may sometimes lead to the wrong root port being elected through the above process. Root protection allows you to manually enforce the root bridge placement in the network.

Enable root protection on interfaces that should not receive higher-priority BPDUs from the root bridge and should not be elected as the root port. These interfaces become designated ports and are typically located on an administrative boundary. If the bridge receives more STP BPDUs on a port that has root protection enabled, that port transitions to a root-prevented STP state (inconsistency state), and the interface is blocked. This blocking prevents a bridge that should not be the root bridge from being elected the root bridge. After the bridge stops receiving more STP BPDUs on the interface with root protection, the interface returns to a listening state, followed by a learning state, and ultimately back to a forwarding state. Recovery back to the forwarding state is automatic.

When root protection is enabled on an interface, it is enabled for all the STP instances on that interface. The interface is blocked only for instances for which it receives more BPDUs. Otherwise, it participates in the spanning-tree topology.

An interface can be configured for either root protection or loop protection, but not for both.

Related Documentation

- Example: Configuring Root Protection to Enforce Root Bridge Placement in Spanning Trees on page 207
- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Understanding MSTP on page 52

- Understanding RSTP on page 53
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54

CHAPTER 4

Unified Forwarding Table

Understanding the Unified Forwarding Table

Using the Unified Forwarding Table to Optimize Address Storage

On QFX5100 and EX4600 switches, you can control the allocation of forwarding table memory available to store the following:

- MAC addresses.
- Layer 3 host entries.
- Longest prefix match (LPM) table entries.

NOTE: **Starting with Junos OS 13.2X51-D15, you can allocate more memory to store prefixes in the range /65 to /127 range.**

This feature gives you the flexibility to configure your switch to match the needs of your particular network environment.

MAC Address and Host Address Memory Allocation

There are several profiles that allocate memory differently for MAC addresses and host addresses. You configure the mix that best meets your needs by choosing the appropriate profile. Table 8 on page 59 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries.

Table 8: Unified Forwarding Table Profiles

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)

Table 8: Unified Forwarding Table Profiles *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K
lpm-profile	32K	16K	8K	8K	8K	4K	4K

Note that all entries in the host table share the same memory space. If the host table stores the maximum number of entries for any given type, the entire shared table is full and is unable to accommodate *any* entries of any other type. As you can see, different entry types occupy different amounts of memory. For example, an IPv6 unicast address occupies twice as much memory as an IPv4 unicast address, and an IPv6 multicast address occupies four times as much memory as an IPv4 unicast address.
Table 9 on page 60 lists various valid combinations that the host table can store if you use the **l2-profile-one** profile. Each row in the table represents a case in which the host table is full and cannot accommodate any more entries. .

Table 9: Example Host Table Combinations Using l2-profile-one

IPv4 unicast	IPv6 unicast	IPv4 multicast (*, G)	IPv4 multicast (S, G)	IPv6 multicast (*, G)	IPv6 multicast (S, G)
16K	0	0	0	0	0
12K	2K	0	0	0	0
12K	0	2K	2K	0	0
8K	4K	0	0	0	0
4K	2K	2K	2K	0	0
0	4K	0	0	1K	1K

LPM Table Memory Allocation

You configure the memory allocation for LPM table entries differently depending on which version of Junos OS you use. To learn how to configure memory allocation for LPM table entries see "Configuring the Unified Forwarding Table" on page 225.

Related Documentation

- Configuring the Unified Forwarding Table on page 225

CHAPTER 5

Q-in-Q Tunneling

Understanding Q-in-Q Tunneling and VLAN Translation

Q-in-Q tunneling and VLAN translation allow service providers to create a Layer 2 Ethernet connection between two customer sites. Providers can segregate different customers' VLAN traffic on a link (for example, if the customers use overlapping VLAN IDs) or bundle different customer VLANs into a single service VLAN. Data centers can use Q-in-Q tunneling and VLAN translation to isolate customer traffic within a single site or to enable customer traffic flows between cloud data centers in different geographic locations.

Q-in-Q tunneling adds a service VLAN tag before the customer's 802.1Q VLAN tags. The Juniper Networks Junos operating system implementation of Q-in-Q tunneling supports the IEEE 802.1ad standard.

All of the VLANs in an implementation can be service VLANs. That is, if the total number of supported VLANs is 4090, all of them can be service VLANs.

This topic describes:

How Q-in-Q Tunneling Works

In Q-in-Q tunneling, as a packet travels from a customer VLAN (C-VLAN) to a service provider's or data center VLAN (S-VLAN), another 802.1Q tag for the appropriate S-VLAN is added before the C-VLAN tag. The C-VLAN tag remains and is transmitted through the network. As the packet leaves the S-VLAN in the downstream direction, the S-VLAN 802.1Q tag is removed.

An interface can be a member of multiple S-VLANs. You can map one C-VLAN to one S-VLAN or multiple C-VLANs to one S-VLAN. C-VLAN and S-VLAN tags use separate

name spaces, so you can have both a C-VLAN 101 and an S-VLAN 101, for example. You can limit the set of accepted customer tags to a range of tags or to discrete values.

When Q-in-Q tunneling is enabled, trunk interfaces are assumed to be part of the service provider or data center network. Access interfaces are assumed to be customer-facing and accept both tagged and untagged frames. When using many-to-one bundling or mapping a specific interface, you must use the **native** option to specify an S-VLAN for untagged and priority tagged packets if you want to accept these packets. (Priority tagged packets have their VLAN ID set to 0, and their priority code point bits might be configured with a CoS value.) If you do not specify an S-VLAN for them, untagged packets are discarded. The **native** option is not available for all-in-one bundling because there is no need to specify untagged and priority tagged packets when all packets are mapped to an S-VLAN.

On QFabric systems only, you can use the **native** option to apply a specified inner tag to packets that ingress as untagged on access interfaces. This functionality is useful if your QFabric system connects to servers that host customer virtual machines that send untagged traffic and each customer's traffic requires its own VLAN while being transported through the QFabric. Instead of using individual VLANs for each customer (which can quickly lead to VLAN exhaustion), you can apply a unique inner (C-VLAN) tag to each customer's traffic and then apply a single outer tag (S-VLAN) tag for transport through the QFabric. This allows you to segregate your customers's traffic while consuming only one QFabric VLAN. Use the **inner-tag** option of the mapping statement to accomplish this.

Q-in-Q tunneling does not affect any class-of-service (CoS) values that are configured on a C-VLAN. These settings are retained in the C-VLAN tag and can be used after a packet leaves an S-VLAN. CoS values are not copied from C-VLAN tags to S-VLAN tags.

Depending on your interface configuration, you might need to adjust the MTU value on your trunk or access ports to accommodate the 4 bytes used for the tag added by Q-in-Q tunneling. For example, if you use the default MTU value of 1514 bytes on your access and trunk ports, you need to make one of the following adjustments:

- Reduce the MTU on the access links by at least 4 bytes so that the frames do not exceed the MTU of the trunk link when S-VLAN tags are added.
- Increase the MTU on the trunk link so that the link can handle the larger frame size.

NOTE: You can configure Q-in-Q tunneling only on access ports (not trunk ports).

How VLAN Translation Works

VLAN translation replaces an incoming C-VLAN tag with an S-VLAN tag instead of adding an additional tag. The C-VLAN tag is therefore lost, so a single-tagged packet is normally untagged when it leaves the S-VLAN (at the other end of the link). If an incoming packet has had Q-in-Q tunneling applied in advance, VLAN translation replaces the outer tag

and the inner tag is retained when the packet leaves the S-VLAN at the other end of the link.

To configure VLAN translation, use the mapping **swap** statement at the **[edit vlans interface]** hierarchy level.

NOTE: You can configure VLAN translation on access ports only. You cannot configure it on trunk ports, and you cannot configure Q-in-Q tunneling on the same access port.

NOTE: VLAN translation is not supported on QFabric systems.

Mapping C-VLANs to S-VLANs

The three ways to map C-VLANs to an S-VLAN are:

- All-in-one bundling—Use the **edit vlans** *s-vlan-name* **dot1q-tunneling** statement without specifying customer VLANs. All packets received on all access interfaces (including untagged packets) are mapped to the S-VLAN.
- Many-to-one bundling—Use the **edit vlans** *s-vlan-name* **dot1q-tunneling customer-vlans** statement to specify which C-VLANs are mapped to the S-VLAN. Use this method when you want a subset of the C-VLANs to be part of the S-VLAN. If you want untagged or priority tagged packets to be mapped to the S-VLAN, use the **native** option with the **customer-vlans** statement. (Priority tagged packets have their VLAN ID set to 0, and their priority code point bits might be configured with a CoS value.)
- Mapping a specific interface—Use the **edit vlans** *s-vlan-name* **interface** *interface-name* **mapping** statement to specify a C-VLAN for a given S-VLAN. This configuration applies to only one interface—not all access interfaces as with all-in-one and many-to-one bundling. If you want untagged or priority tagged packets to be mapped to the S-VLAN, use the **native** option with the **customer-vlans** statement.

  This method has two options: swap and push. With the push option, a packet retains its tag and an additional VLAN tag is added. With the swap option, the incoming tag is replaced with an S-VLAN tag. (This is VLAN translation.)

  - You can configure multiple push rules for a given S-VLAN and interface. That is, you can configure an interface so that the same S-VLAN tag is added to packets arriving from multiple C-VLANs.
  - You can configure only one swap rule for a given S-VLAN and interface.

  This functionality is typically used to keep traffic from different customers separate or to provide individualized treatment for traffic on a certain interface.

If you configure multiple methods, the switch prioritizes the mappings in the following order:

1. Specific interface mapping

2. Many-to-one bundling
3. All-in-one bundling

You cannot configure overlapping rules for the same C-VLAN using the same mapping method. For example, you cannot use many-to one bundling to map C-VLAN 100 to two different S-VLANs.

Routed VLAN Interfaces on Q-in-Q VLANs

Routed VLAN interfaces (RVIs) are supported on Q-in-Q VLANs. Routing is based on the S-VLAN, and the original C-VLAN tag is dropped when a packet leaves the VLAN that it originated in. Outgoing routed packets retain any S-VLAN tag only when exiting a trunk interface—S-VLAN tags are dropped when traffic exits an access interface.

Constraints for Q-in-Q Tunneling and VLAN Translation

Be aware of the following constraints when configuring Q-in-Q tunneling and VLAN translation:

- Most access port security features are not supported with Q-in-Q tunneling and VLAN translation.
- Configuring Q-in-Q tunneling and VLAN translation on the same port is not supported.
- You can configure at most one VLAN translation for a given VLAN and interface. For example, you can create no more than one translation for VLAN 100 on interface xe-0/0/0.
- The combined total of VLANs and rules for Q-in-Q tunneling and VLAN translation cannot exceed 6000. For example, you can configure and commit 4000 VLANs and 2000 rules for Q-in-Q tunneling and VLAN translation. However, you cannot configure 4000 VLANs and 2500 rules for Q-in-Q tunneling and VLAN translation. If you try to commit a configuration that exceeds the limit, you see CLI and syslog errors that inform you about the problem.
- MAC addresses are learned from S-VLANs, not C-VLANs.
- Broadcast, unknown unicast, and multicast traffic is forwarded to all members in the S-VLAN.
- The following features are not supported with Q-in-Q tunneling:
  - DHCP relay
  - Fibre Channel over Ethernet
  - IP Source Guard
- The following features are not supported with VLAN translation:
  - Fibre Channel over Ethernet
  - Firewall filter applied to a port or VLAN in the output direction
  - Private VLANs

- VLAN Spanning Tree Protocol
- Reflective relay

Related Documentation

- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151
- *Troubleshooting Q-in-Q and VLAN Translation Configuration*
- **mapping on page 308**
- *mtu*

CHAPTER 6

Proxy ARP

Understanding Proxy ARP

You can configure proxy Address Resolution Protocol (ARP) to enable the switch to respond to ARP queries for network addresses by offering its own Ethernet media access control (MAC) address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

Proxy ARP is useful in situations where hosts are on different physical networks and you do not want to use subnet masking. Because ARP broadcasts are not propagated between hosts on different physical networks, hosts will not receive a response to their ARP request if the destination is on a different subnet. Enabling the switch to act as an ARP proxy allows the hosts to transparently communicate with each other through the switch. Proxy ARP can help hosts on a subnet reach remote subnets without your having to configure routing or a default gateway.

What Is ARP?

Ethernet LANs use ARP to map Ethernet MAC addresses to IP addresses. Each device maintains a cache containing a mapping of MAC addresses to IP addresses. The switch maintains this mapping in a cache that it consults when forwarding packets to network devices. If the ARP cache does not contain an entry for the destination device, the host (the DHCP client) broadcasts an ARP request for that device's address and stores the response in the cache.

Proxy ARP Overview

When proxy ARP is enabled, if the switch receives an ARP request for which it has a route to the target (destination) IP address, the switch responds by sending a proxy ARP reply packet containing its own MAC address. The host that sent the ARP request then sends its packets to the switch, which forwards them to the intended host.

NOTE: For security reasons, the source address in an ARP request must be on the same subnet as the interface on which the ARP request is received.

You can configure proxy ARP for each interface. You can also configure proxy ARP for a VLAN by using a routed VLAN interface (RVI).

Two modes of proxy ARP are supported:, restricted and unrestricted. Both modes require that the switch have an active route to the destination address of the ARP request.

- Restricted—The switch responds to ARP requests in which the physical networks of the source and target are different and does not respond if the source and target IP addresses are on the same subnet. In this mode, hosts on the same subnet communicate without proxy ARP. We recommend that you use this mode on the switch.
- Unrestricted—The switch responds to all ARP requests for which it has a route to the destination. This is the default mode (because it is the default mode in Juniper Networks Junos operating system (Junos OS) configurations other than those on the switch). We recommend using restricted mode on the switch.

Best Practices for Proxy ARP

We recommend these best practices for configuring proxy ARP on the switches:

- Set proxy ARP to restricted mode.
- Use restricted mode when configuring proxy ARP on RVIs.
- If you set proxy ARP to unrestricted, disable gratuitous ARP requests on each interface enabled for proxy ARP.

Related Documentation

- Configuring Proxy ARP on page 251
- proxy-arp on page 302

PART 2

Configuration

CHAPTER 7

Configuration Examples

Example: Configuring Automatic VLAN Administration Using MVRP

As the numbers of servers and VLANs attached to a QFabric systems increase, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple redundant server Node group devices becomes increasingly difficult. To partially automate VLAN administration, you can enable Multiple VLAN Registration Protocol (MVRP) on your QFabric system. If your QFabric system connects to servers that host many virtual machines that require their own VLANs, using MVRP can save you the time and effort that would be required to manually configure and administer the VLANs on the interfaces that connect to the servers. For example, if a virtual machine moves between servers—and therefore connects to a different redundant server Node group interface—MVRP can configure the appropriate VLAN membership on the new server Node group interface.

NOTE: **Only trunk interfaces can be enabled for MVRP.**

This example describes how to configure MVRP on a QFabric system.

Requirements

This example uses the following hardware and software components:

- One QFabric system
- Junos OS Release 13.1 for the QFX Series

Overview and Topology

MVRP ensures that the VLAN membership information on the trunk interface is updated as the switch's access interfaces become active or inactive in the configured VLANs in a static or dynamic VLAN creation setup.

You do not need to explicitly bind a VLAN to the trunk interface. When MVRP is enabled, the trunk interface advertises all the VLANs that are active (bound to access interfaces) on that switch. An MVRP-enabled trunk interface does not advertise VLANs that have been configured on the switch but that are not currently bound to an access interface. Thus, MVRP provides the benefit of reducing network overhead—by limiting the scope of broadcast, unknown unicast, and multicast (BUM) traffic to interested devices only.

When VLAN access interfaces become active or inactive, MVRP ensures that the updated information is advertised on the trunk interface. Thus, in this example, distribution Switch C does not forward traffic to inactive VLANs.

A redundant server Node group device is connected to a server that hosts virtual machines for three customers, each of which requires its own VLAN.

- **customer-1**: VLAN ID 100
- **customer-2**: VLAN ID 200
- **customer-3**: VLAN ID 300

Table 10 on page 72 explains the components of the example topology.

Table 10: Components of the Example Topology

Settings	Settings
Hardware	• Redundant server Node group device • Network Node group device
VLAN names and IDs	• **customer-1**, VLAN ID (tag)**100** • **customer-2**, VLAN ID (tag)**200** • **customer-3**, VLAN ID (tag)**300**
Interfaces	Redundant server Node group device interfaces: • **RSNG:xe-0/1/1**—Uplink to interconnect device • **RSNG:xe-0/0/1**—Server-facing interface Network Node group device interface: • **NNG:xe-0/0/1**—Uplink to interconnect device

Configuring VLANs and Network Node Group Interfaces

To configure VLANs, bind the VLANs to the server-facing trunk interface, and enable MVRP on the trunk interface of the network Node group device, perform these tasks:

CLI Quick Configuration

To quickly configure VLANs on the QFabric system, assign VLAN membership to the uplink port on the network Node group device, and configure the uplink port to be trunk:

```
[edit]
set vlans customer-1 vlan-id 100
set vlans customer-2 vlan-id 200
set vlans customer-3 vlan-id 300
set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching vlan members [customer-1
customer-2 customer-3]
```

NOTE: As recommended as a best practice, default MVRP timers are used in this example, so they are not configured. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms for the leaveall timer. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Step-by-Step Procedure

To create the VLANs and configure the network Node group device for MVRP, follow these steps. Note that you are creating VLANs for the entire QFabric system, so you do not need to create them on specific QFabric devices.

1. Configure the VLAN for customer 1:

   ```
   [edit]
   user@qfabric# set vlans customer-1 vlan-id 100
   ```

2. Configure the VLAN for customer 2:

   ```
   [edit]
   user@qfabric# set vlans customer-2 vlan–id 200
   ```

3. Configure the VLAN for customer 3:

   ```
   [edit]
   user@qfabric# set vlans customer-3 vlan–id 300
   ```

4. Configure an uplink interface (one that connects to an interconnect device) to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching port-mode
   trunk
   ```

5. Configure the uplink interface to be a member of all three VLANs:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/0/1 unit 1 family ethernet-switching vlan members
   [customer-1 customer-2 customer-3]
   ```

NOTE: If you want the uplink interface to be a member of all the VLANs in the QFabric system, you can enter all instead of specifying the individual VLANs.

Results

Check the results of the configuration on the network Node group device:

```
[edit]
user@qfabric# show interfaces NNG:xe-0/0/1.0
family ethernet-switching {
   port-mode trunk;
   vlan {
      members customer-1 customer-2 customer-3;
   }
}

[edit]
user@qfabric# show vlans
customer-1 {
   vlan-id 100;
}
customer-2 {
   vlan-id 200;
}
customer-3 {
   vlan-id 300;
}
```

Configuring the Redundant Server Node Group

CLI Quick Configuration

To quickly configure the redundant server Node group device for MVRP:

```
[edit]
set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching vlan members [customer-1
customer-2 customer-3]
set interfaces RSNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set protocols mvrp interface RSNG:xe-0/0/1.0 passive
```

Step-by-Step Procedure

To configure the redundant server Node group device, follow these steps. Note that you do not need to configure the VLANs on the server-facing interface (RSNG:xe-0/0/1), but you do need to configure the VLANs on the uplink interface. Also notice that in this example you configure the server-facing interface to be passive, which means that it will not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from the server.

1. Configure an uplink interface (one that connects to the interconnect device) to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching port-mode
   trunk
   ```

2. Configure the uplink interface to be a member of all three VLANs:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/1/1 unit 0 family ethernet-switching vlan members
   [customer-1 customer-2 customer-3]
   ```

3. Configure an interface that connects to the server that hosts multiple virtual machines to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces RSNG:xe-0/0/1 unit 0 family ethernet-switching port-mode
   trunk
   ```

4. Enable MVRP on the server-facing trunk interface and configure it to be passive:

```
[edit]
user@qfabric# set protocols mvrp interface RSNG:xe-0/0/1.0 passive
```

Results Check the results of the configuration for the redundant server Node group:

```
[edit]
user@qfabric# show interfaces RSNG:xe-0/0/1.0
family ethernet-switching {
   port-mode trunk;
}

[edit]
user@qfabric# show interfaces RSNG:xe-0/1/1.0
family ethernet-switching {
   port-mode trunk;
   }
passive
}

[edit]
user@qfabric# show protocols mvrp
interface RSNG:xe-0/0/1.0;
```

Verification

To confirm that the configuration is updating VLAN membership, perform these tasks:

- Verifying That MVRP Is Enabled On The QFabric System on page 75

Verifying That MVRP Is Enabled On The QFabric System

Purpose Verify that MVRP is enabled on the appropriate interfaces

Action Show the MVRP configuration:

```
user@qfabric> show mvrp

MVRP configuration
MVRP status                 : Enabled

MVRP timers (ms):
Interface          Join   Leave       LeaveAll
--------------     -----  --------    -----------
NNG:xe-0/0/1.0     200     1000       10000
RSNG:xe-0/0/1.0    200     1000       10000
RSNG:xe-0/1/1.0    200     1000       10000


Interface           Status        Registration Mode
--------------  --------      -----------------
NNG:xe-0/0/1.0      Enabled       Normal
RSNG:xe-0/1/1.0     Enabled       Normal
RSNG:xe-0/0/1.0     Enabled       Passive
```

Meaning The results show that MVRP is enabled on the appropriate network Node group and redundant server Node group interfaces and that the default timers are used.

Related Documentation

- Understanding Multiple VLAN Registration Protocol (MVRP) on page 47

Example: Connecting an Access Switch to a Distribution Switch

In large local area networks (LANs), you commonly need to aggregate traffic from a number of access switches into a distribution switch.

This example describes how to connect an access switch to a distribution switch:

Requirements

This example uses the following hardware and software components:

- For the distribution switch, one EX 4200-24F switch. This model is designed to be used as a distribution switch for aggregation or collapsed core network topologies and in space-constrained data centers. It has twenty-four 1-Gigabit Ethernet fiber SFP ports and an EX-UM-2XFP uplink module with two 10-Gigabit Ethernet XFP ports.
- For the access switch, one EX 3200-24P, which has twenty-four 1-Gigabit Ethernet ports, all of which support Power over Ethernet (PoE), and an uplink module with four 1-Gigabit Ethernet ports.
- Junos OS Release 11.1 or later for the QFX Series

Overview and Topology

In a large office that is spread across several floors or buildings, or in a data center, you commonly aggregate traffic from a number of access switches into a distribution switch. This configuration example shows a simple topology to illustrate how to connect a single access switch to a distribution switch.

In the topology, the LAN is segmented into two VLANs, one for the sales department and the second for the support team. One 1-Gigabit Ethernet port on the access switch's uplink module connects to the distribution switch, to one 1-Gigabit Ethernet port on the distribution switch.

Table 11 on page 76 explains the components of the example topology. The example shows how to configure one of the three access switches. The other access switches could be configured in the same manner.

Table 11: Components of the Topology for Connecting an Access Switch to a Distribution Switch

Property	Settings

Table 11: Components of the Topology for Connecting an Access Switch to a Distribution Switch *(continued)*

Access switch hardware	EX 3200-24P, 24 1-Gigabit Ethernet ports, all PoE-enabled (**ge-0/0/0** through **ge-0/0/23**); one 4-port 1–Gigabit Ethernet uplink module (EX-UM-4SFP)
Distribution switch hardware	EX 4200-24F, 24 1-Gigabit Ethernet fiber SPF ports (**ge-0/0/0** through **ge-0/0/23**); one 2–port 10–Gigabit Ethernet XFP uplink module (EX-UM-4SFP)
VLAN names and tag IDs	**sales**, tag **100support**, tag **200**
VLAN subnets	**sales**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**)**support**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Trunk port interfaces	On the access switch: **ge-0/1/0**On the distribution switch: **ge-0/0/0**
Access port interfaces in VLAN **sales** (on access switch)	Avaya IP telephones: **ge-0/0/3** through **ge-0/0/19**Wireless access points: **ge-0/0/0** and **ge-0/0/1**Printers: **ge-0/0/22** and **ge-0/0/23**File servers: **ge-0/0/20** and **ge-0/0/21**
Access port interfaces in VLAN **support** (on access switch)	Avaya IP telephones: **ge-0/0/25** through **ge-0/0/43**Wireless access points: **ge-0/0/24**Printers: **ge-0/0/44** and **ge-0/0/45**File servers: **ge-0/0/46** and **ge-0/0/47**
Unused interfaces on access switch	**ge-0/0/2** and **ge-0/0/25**

Configuring the Access Switch

To configure the access switch:

CLI Quick Configuration

To quickly configure the access switch, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces ge-0/0/0 unit 0 description “Sales Wireless access point port”
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/3 unit 0 description “Sales phone port”
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/22 unit 0 description “Sales printer port”
set interfaces ge-0/0/22 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/20 unit 0 description “Sales file server port”
set interfaces ge-0/0/20 unit 0 family ethernet-switching vlan members sales
set interfaces ge-0/0/24 unit 0 description “Support wireless access point port”
set interfaces ge-0/0/24 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/0/26 unit 0 description “Support phone port”
set interfaces ge-0/0/26 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/0/44 unit 0 description “Support printer port”
set interfaces ge-0/0/44 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/0/46 unit 0 description “Support file server port”
set interfaces ge-0/0/46 unit 0 family ethernet-switching vlan members support
set interfaces ge-0/1/0 unit 0 description “Uplink module port connection to distribution switch”
set interfaces ge-0/1/0 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/1/0 unit 0 family ethernet-switching native-vlan-id 1
set interfaces ge-0/1/0 unit 0 family ethernet switching vlan members [sales support]
set interfaces vlan unit 0 family inet address 192.0.2.1/25
set interfaces vlan unit 1 family inet address 192.0.2.129/25
```

```
set vlans sales interface ge-0/0/0.0
set vlans sales interface ge-0/0/3.0
set vlans sales interface ge-0/0/22.0
set vlans sales interface ge-0/0/20.0
set vlans sales l3-interface vlan.0
set vlans sales vlan-id 100
set vlans sales vlan-description “Sales VLAN”
set vlans support interface ge-0/0/24.0
set vlans support interface ge-0/0/26.0
set vlans support interface ge-0/0/44.0
set vlans support interface ge-0/0/46.0
set vlans support vlan-id 200
set vlans support l3–interface vlan.1
set vlans support vlan-description “Support VLAN”
```

Step-by-Step Procedure

To configure the access switch:

1. Configure the 1-Gigabit Ethernet interface on the uplink module to be the trunk port that connects to the distribution switch:

   ```
   [edit interfaces ge-0/1/0 unit 0]user@access-switch# setdescription “Uplink module port connection to distribution switch”user@access-switch# set ethernet-switching port-mode trunk
   ```

2. Specify the VLANs to be aggregated on the trunk port:

   ```
   [edit interfaces ge-0/1/0 unit 0]user@access-switch# set ethernet-switching vlanmembers [ sales support ]
   ```

3. Configure the VLAN ID to use for packets that are received with no dot1q tag (untagged packets):

   ```
   [edit interfaces ge-0/1/0 unit 0]user@access-switch# set ethernet-switching native-vlan-id 1
   ```

4. Configure the sales VLAN:

   ```
   [edit vlans sales]user@access-switch# set vlan-description “Sales VLAN”user@access-switch# set vlan-id (VLANs) 100user@access-switch# set l3-interface (VLAN) vlan.0
   ```

5. Configure the support VLAN:

   ```
   [edit vlans support]user@access-switch# set vlan-description “Support VLAN”user@access-switch# set vlan-id (VLANs) 200user@access-switch# set l3-interface (VLAN) vlan.1
   ```

6. Create the subnet for the sales broadcast domain:

   ```
   [edit interfaces]user@access-switch# set vlan unit 0 family inet address 192.0.2.1/25
   ```

7. Create the subnet for the support broadcast domain:

   ```
   [edit interfaces]user@access-switch# set vlan unit 1 family inet address 192.0.2.129/25
   ```

8. Configure the interfaces in the sales VLAN:

   ```
   [edit interfaces]user@access-switch# set ge-0/0/0 unit 0 description “Sales wireless access point port”user@access-switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members salesuser@access-switch# set ge-0/0/3 unit 0 description “Sales phone port”user@access-switch# set ge-0/0/3 unit 0 family ethernet-switching vlan members salesuser@access-switch# set ge-0/0/20 unit 0 description “Sales file server port”user@access–switch# set ge-0/0/20 unit 0 family ethernet-switching vlan members salesuser@access-switch# set ge-0/0/22 unit 0 description “Sales printer port”user@access-switch# set ge-0/0/22 unit 0 family ethernet-switching vlan members sales
   ```

9. Configure the interfaces in the support VLAN:

```
[edit interfaces]user@access-switch# set ge-0/0/24 unit 0 description “Support
wireless access point port”user@access-switch# set ge-0/0/24 unit 0 family
ethernet-switching vlan members supportuser@access-switch# set ge-0/0/26 unit 0
description “Support phone port”user@access-switch# set ge-0/0/26 unit 0 family
ethernet-switching vlan members supportuser@access-switch# set ge-0/0/44 unit 0
description “Support printer port”user@access-switch# set ge-0/0/44 unit 0 family
ethernet-switching vlan members supportuser@access-switch# set ge-0/0/46 unit 0
description “Support file server port”user@access-switch# set ge-0/0/46 unit 0 family
ethernet-switching vlan members support
```

10. Configure descriptions and VLAN tag IDs for the sales and support VLANs:

```
[edit vlans]user@access-switch# set sales vlan-description “Sales
VLAN”user@access-switch# set sales vlan-id 100user@access-switch# set support
vlan-description “Support VLAN”user@access-switch# set support vlan-id 200
```

11. To route traffic between the sales and support VLANs and associate a Layer 3 interface with each VLAN:

```
[edit vlans]user@access-switch# set sales l3-interface vlan.0user@access-switch#
set support l3-interface vlan.1
```

Results Display the results of the configuration:

```
user@access-switch> show
interfaces {
  ge-0/0/0 {
    unit 0 {
      description “Sales wireless access point port”;
      family ethernet-switching {
        vlan members sales;
      }
    }
  }
  ge-0/0/3 {
    unit 0 {
      description “Sales phone port”;
        family ethernet-switching {
        vlan members sales;
      }
    }
  }
  ge-0/0/20 {
    unit 0 {
      description “Sales file server port”;
        family ethernet-switching {
        vlan members sales;
      }
    }
  }
  ge-0/0/22 {
    unit 0 {
      description “Sales printer port”;
        family ethernet-switching {
        vlan members sales;
      }
    }
  }
  ge-0/0/24 {
```

```
    unit 0 {
      description “Support wireless access point port”;
        family ethernet-switching {
        vlan members support;
      }
    }
  }
  ge-0/0/26 {
    unit 0 {
      description “Support phone port”;
        family ethernet-switching {
        vlan members support;
      }
    }
  }
  ge-0/0/44 {
    unit 0 {
      description “Support printer port”;
        family ethernet-switching {
        vlan members sales;
      }
    }
  }
  ge-0/0/46 {
    unit 0 {
      description “Support file server port”;
        family ethernet-switching {
        vlan members support;
      }
    }
  }
  ge-0/1/0 {
    unit 0 {
      description “Uplink module port connection to distribution switch”;
        family ethernet-switching {
        port-mode trunk;
          vlan members [ sales support ];
        native-vlan-id 1;
      }
    }
  }
  vlan {
    unit 0 {
      family inet address 192.0.2.1/25;
    }
    unit 1 {
      family inet address 192.0.2.129/25;
    }
  }
}
vlans {
  sales {
    vlan-id 100;
    vlan-description “Sales VLAN”;
    l3-interface vlan.0;
  }
```

```
    support {
      vlan-id 200;
      vlan-description “Support VLAN”;
      l3-interface vlan.1;
    }
  }
```

TIP: To quickly configure the distribution switch, issue the load merge terminal command, then copy the hierarchy and paste it into the switch terminal window.

Configuring the Distribution Switch

To configure the distribution switch:

CLI Quick Configuration

To quickly configure the distribution switch, copy the following commands and paste them into the switch terminal window:

```
set interfaces ge-0/0/0 description “Connection to access switch”
set interfaces ge-0/0/0 ethernet-switching port-mode trunk
set interfaces ge-0/0/0 ethernet-switching vlan members [ sales support ]
set interfaces ge-0/0/0 ethernet-switching native-vlan-id 1
set interfaces vlan unit 0 family inet address 192.0.2.2/25
set interfaces vlan unit 1 family inet address 192.0.2.130/25
set vlans sales vlan-description “Sales VLAN”
set vlans sales vlan-id 100
set vlans sales l3-interface vlan.0
set vlans support vlan-description “Support VLAN”
set vlans support vlan-id 200
set vlans support l3-interface vlan.1
```

Step-by-Step Procedure

To configure the distribution switch:

1. Configure the interface on the switch to be the trunk port that connects to the access switch:

   ```
   [edit interfaces ge-0/0/0 unit 0]user@distribution-switch# set description
   “Connection to access switch”user@distribution-switch# set ethernet-switching
   port-mode trunk
   ```

2. Specify the VLANs to be aggregated on the trunk port:

   ```
   [edit interfaces ge-0/0/0 unit 0]user@distribution-switch# set ethernet-switching
   vlanmembers [ sales support ]
   ```

3. Configure the VLAN ID to use for packets that are received with no dot1q tag (untagged packets):

   ```
   [edit interfaces]user@distribution-switch# set ge-0/0/0 ethernet-switching
   native-vlan-id 1
   ```

4. Configure the sales VLAN:

   ```
   [edit vlans sales]user@distribution-switch# set vlan-description “Sales
   VLAN”user@distribution-switch# set vlan-id (VLANs) 100user@distribution-switch#
   set l3-interface (VLAN) vlan.0
   ```

5. Configure the support VLAN:

```
[edit vlans support]user@distribution-switch# set vlan-description “Support
VLAN”user@distribution-switch# set vlan-id (VLANs) 200user@distribution-switch#
set l3-interface (VLAN) vlan.1
```

6. Create the subnet for the sales broadcast domain:

```
[edit interfaces]user@distribution-switch# set vlan unit 0 family inet address
192.0.2.2/25
```

7. Create the subnet for the support broadcast domain:

```
[edit interfaces] user@distribution-switch# set vlan unit 1 family inet address
192.0.2.130/25
```

Results Display the results of the configuration:

```
user@distribution-switch> show
interfaces {
  ge-0/0/0 {
    description “Connection to access switch”;
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        vlan members [ sales support ];
        native-vlan-id 1;
      }
    }
  }
  vlan {
    unit 0 {
      family inet address 192.0.2.2/25;
    }
    unit 1 {
      family inet address 192.0.2.130/25;
    }
  }
}
vlans {
  sales {
    vlan-id 100;
    vlan-description “Sales VLAN”;
    l3-interface vlan.0;
  }
  support {
    vlan-id 200;
    vlan-description “Support VLAN”;
    l3-interface vlan.1;
  }
}
```

TIP: To quickly configure the distribution switch, issue the load merge terminal command, then copy the hierarchy and paste it into the switch terminal window.

Verification

To confirm that the configuration is working properly, perform these tasks:

Verifying the VLAN Members and Interfaces on the Access Switch

Purpose Verify that the **sales** and **support** have been created on the switch.

Action List all VLANs configured on the switch:

```
user@switch> show vlans

Name            Tag     Interfaces
default
                        ge-0/0/1.0, ge-0/0/2.0, ge-0/0/4.0, ge-0/0/5.0,
                        ge-0/0/6.0, ge-0/0/7.0, ge-0/0/8.0*, ge-0/0/9.0,

                        ge-0/0/10.0, ge-0/0/11.0*, ge-0/0/12.0, ge-0/0/13.0,
                        ge-0/0/14.0, ge-0/0/15.0, ge-0/0/16.0, ge-0/0/17.0,
                        ge-0/0/18.0, ge-0/0/19.0*,ge-0/0/21.0, ge-0/0/23.0,
                        ge-0/0/25.0, ge-0/0/27.0*,ge-0/0/28.0, ge-0/0/29.0,
                        ge-0/0/30.0, ge-0/0/31.0*,ge-0/0/32.0, ge-0/0/33.0,
                        ge-0/0/34.0, ge-0/0/35.0*,ge-0/0/36.0, ge-0/0/37.0,
                        ge-0/0/38.0, ge-0/0/39.0*,ge-0/0/40.0, ge-0/0/41.0,
                        ge-0/0/42.0, ge-0/0/43.0*,ge-0/0/45.0, ge-0/0/47.0,
                        ge-0/1/1.0*, ge-0/1/2.0*, ge-0/1/3.0*

sales           100
                        ge-0/0/0.0*, ge-0/0/3.0, ge-0/0/20.0, ge-0/0/22.0,
                        ge-0/1/0.0*,

support         200
                        ge-0/0/24.0*, ge-0/0/26.0, ge-0/0/44.0, ge-0/0/46.0,

mgmt
                        me0.0*
```

Meaning The output shows the **sales** and **support** VLANs and the interfaces associated with them.

Verifying the VLAN Members and Interfaces on the Distribution Switch

Purpose Verify that the **sales** and **support** have been created on the switch.

Action List all VLANs configured on the switch:

```
user@switch> show vlans

Name            Tag     Interfaces
default
                        ge-0/0/1.0, ge-0/0/2.0, ge-0/0/3.0, ge-0/0/4.0,
                        ge-0/0/5.0, ge-0/0/6.0, ge-0/0/7.0*, ge-0/0/8.0,

                        ge-0/0/9.0, ge-0/0/10.0*, ge-0/0/11.0, ge-0/0/12.0,
                        ge-0/0/13.0, ge-0/0/14.0, ge-0/0/15.0, ge-0/0/16.0,
                        ge-0/0/17.0, ge-0/0/18.0*, ge-0/0/19.0, ge-0/0/20.0,
                        ge-0/0/21.0, ge-0/0/22.0*, ge-0/0/23.0, ge-0/1/1.0*,
                        ge-0/1/2.0*, ge-0/1/3.0*

sales           100
                        ge-0/0/0.0*

support         200
                        ge-0/0/0.0*

mgmt
                        me0.0*
```

Meaning The output shows the **sales** and **support** VLANs associated to interface **ge-0/0/0.0**. Interface **ge-0/0/0.0** is the trunk interface connected to the access switch.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Understanding Bridging*

CHAPTER 8

Bridging Configuration Examples

Example: Setting Up Basic Bridging and a VLAN on the QFX Series

The QFX Series products use bridging and virtual LANs (VLANs) to connect network devices—storage devices, file servers, and other LAN components—in a LAN and to segment the LAN into smaller bridging domains.

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on a switch. Each VLAN is a collection of network nodes. When you use VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN, and only frames not destined for the local VLAN are forwarded to other broadcast domains. VLANs thus limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN.

This example describes how to configure basic bridging and VLANs for the QFX Series:

Requirements

This example uses the following software and hardware components:

- Junos OS Release 11.1 or later for the QFX Series
- A configured and provisioned QFX Series product

Overview and Topology

To use a switch to connect network devices on a LAN, you must at a minimum configure bridging and VLANs. By default, bridging is enabled on all switch interfaces, all interfaces are in access mode, and all interfaces belong to a VLAN called **employee-vlan**, which is

automatically configured. When you plug in access devices—such as desktop computers, file servers, and printers—they are joined immediately into the **employee-vlan** VLAN, and the LAN is up and running.

The topology used in this example consists of a single QFX3500 switch, with a total of 48 10-Gbps Ethernet ports. (For the purposes of this example, the QSFP+ ports Q0-Q3, which are ports xe-0/1/0 through xe-0/1/15, are excluded.) You use the ports to connect devices that have their own power sources. Table 1 details the topology used in this configuration example.

Table 12: Components of the Basic Bridging Configuration Topology

Property	Settings
Switch hardware	QFX3500 switch, with 48 10-Gbps Ethernet ports
VLAN name	**employee-vlan**
VLAN ID	10
Connections to file servers	**xe-0/0/17** and **xe-0/0/18**
Direct connections to desktop PCs and laptops	**xe-0/0/0** through **xe-0/0/16**
Connections to integrated printer/fax/copier machines	**xe-0/0/19** through **xe-0/0/40**
Unused ports	**xe-0/0/41** through **xe-0/0/47**

Configuration

CLI Quick Configuration

To quickly configure a VLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans employee-vlan vlan-id 10
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/15 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/16 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
```

```
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/22 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/23 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/25 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/27 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/28 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/29 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/30 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/31 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/32 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/33 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/34 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/35 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/36 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/37 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/38 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/39 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces xe-0/0/40 unit 0 family ethernet-switching vlan members employee-vlan
```

Step-by-Step Procedure

To set up basic bridging and a VLAN:

1. Create a VLAN named employee-vlan and specify the VLAN ID of 10 for it:

   ```
   [edit vlans]
   user@switch# set employee-vlan vlan-id 10
   ```

2. Assign interfaces xe-0/0/0 through xe-0/0/40 to the employee-vlan VLAN:

   ```
   [edit interface]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/13 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/14 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/15 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/16 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/21 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/22 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/23 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/24 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/25 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/26 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/27 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set xe-0/0/28 unit 0 family ethernet-switching vlan members employee-vlan
   ```

```
user@switch# set xe-0/0/29 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/30 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/31 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/32 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/33 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/34 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/35 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/36 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/37 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/38 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/39 unit 0 family ethernet-switching vlan members employee-vlan
user@switch# set xe-0/0/40 unit 0 family ethernet-switching vlan members employee-vlan
```

3. Connect the two file servers to ports xe-0/0/17 and xe-0/0/18.
4. Connect the desktop PCs and laptops to ports xe-0/0/0 through xe-0/0/16.
5. Connect the integrated printer/fax/copier machines to ports xe-0/0/19 through xe-0/0/40.

Results Check the results of the configuration:

```
user@switch> show configuration
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/4 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
            family ethernet-switching {
```

```
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/7 {
    unit 0 {
        family ethernet-switching {
             vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/8 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/9 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/10 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/11 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/12 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
```

```
xe-0/0/13 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/14 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/15 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/16 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/17 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/18 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/19 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
```

```
    }
}
xe-0/0/20 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/21 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/22 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/23 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/24 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/25 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/26 {
    unit 0 {
        family ethernet-switching {
                    vlan {
                members employee-vlan;
```

```
            }
        }
    }
}
xe-0/0/27 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/28 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/29 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/30 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/31 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/32 {
    unit 0 {
        family ethernet-switching {
                   vlan {
                members employee-vlan;
            }
        }
    }
}
xe-0/0/33 {
    unit 0 {
        family ethernet-switching {
```

```
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/34 {
        unit 0 {
            family ethernet-switching {
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/35 {
        unit 0 {
            family ethernet-switching {
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/36 {
        unit 0 {
            family ethernet-switching {
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/37 {
        unit 0 {
            family ethernet-switching {
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/38 {
        unit 0 {
            family ethernet-switching {
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/39 {
        unit 0 {
            family ethernet-switching {
                    vlan {
                    members employee-vlan;
                }
            }
        }
    }
    xe-0/0/40 {
```

```
        unit 0 {
            family ethernet-switching {
                       vlan {
                      members employee-vlan;
                   }
            }
        }
    }
```

Verification

To verify that switching is operational and that **employee-vlan** has been created, perform these tasks:

- Verifying That the VLAN Has Been Created on page 95
- Verifying That Interfaces Are Associated with the Proper VLANs on page 96

Verifying That the VLAN Has Been Created

Purpose Verify that the VLAN named **employee-vlan** has been created on the switch.

Action List all VLANs configured on the switch:

```
user@switch> show vlans
Routing instance        VLAN name             Tag         Interfaces
default-switch          employee-vlan          10
                                                           xe-0/0/0.0
                                                           xe-0/0/1.0
                                                           xe-0/0/2.0
                                                           xe-0/0/3.0
                                                           xe-0/0/4.0
                                                           xe-0/0/5.0
                                                           xe-0/0/6.0
                                                           xe-0/0/7.0
                                                           xe-0/0/8.0
                                                           xe-0/0/9.0
                                                           xe-0/0/10.0
                                                           xe-0/0/11.0
                                                           xe-0/0/12.0
                                                           xe-0/0/13.0
                                                           xe-0/0/14.0
                                                           xe-0/0/15.0
                                                           xe-0/0/16.0
                                                           xe-0/0/17.0
                                                           xe-0/0/18.0
                                                           xe-0/0/19.0
                                                           xe-0/0/20.0
                                                           xe-0/0/21.0
                                                           xe-0/0/22.0
                                                           xe-0/0/23.0
                                                           xe-0/0/24.0
                                                           xe-0/0/25.0
                                                           xe-0/0/26.0
                                                           xe-0/0/27.0
                                                           xe-0/0/28.0
                                                           xe-0/0/29.0
                                                           xe-0/0/30.0
                                                           xe-0/0/31.0
                                                           xe-0/0/32.0
                                                           xe-0/0/33.0
                                                           xe-0/0/34.0
                                                           xe-0/0/35.0
                                                           xe-0/0/36.0
                                                           xe-0/0/37.0
                                                           xe-0/0/38.0
                                                           xe-0/0/39.0
                                                           xe-0/0/40.0
...
```

Meaning The **show vlans** command lists the VLANs configured on the switch. This output shows that the VLAN **employee-vlan** has been created.

Verifying That Interfaces Are Associated with the Proper VLANs

Purpose Verify that Ethernet switching is enabled on switch interfaces and that all interfaces are included in the VLAN.

Action List all interfaces on which switching is enabled:

```
user@switch> show ethernet-switching interfaces
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/0.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/1.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/2.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/3.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/4.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/5.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/6.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
```

```
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/7.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/8.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/9.0                    65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/10.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/11.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/12.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/13.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/14.0                   65535                                 untagged
```

```
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/15.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/16.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/17.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/18.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/19.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/20.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/21.0                   65535                                 untagged
            employee-vlan 10
                                65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
```

```
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/22.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/23.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/24.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/25.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/26.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/27.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/28.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/29.0                   65535                                untagged
             employee-vlan 10
                              65535    Discarding
```

```
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/30.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/31.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/32.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/33.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/34.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/35.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/36.0                   65535                                 untagged
             employee-vlan 10
                              65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
```

```
xe-0/0/37.0                  65535                                  untagged
             employee-vlan 10
                             65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/38.0                  65535                                  untagged
             employee-vlan 10
                             65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/39.0                  65535                                  untagged
             employee-vlan 10
                             65535    Discarding
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down )
Logical      Vlan       TAG   MAC      STP          Logical         Tagging
interface    members          limit    state        interface flags
xe-0/0/40.0                  65535                                  untagged
             employee-vlan 10
                             65535    Discarding
...
```

Meaning The **show ethernet-switching interfaces** command lists all interfaces on which switching is enabled (in the **Logical interface** column), along with the VLANs that are active on the interfaces (in the **VLAN members** column). The output in this example shows all the connected interfaces, xe-0/0/0 through xe-0/0/40, are all part of VLAN **employee-vlan**. Notice that the interfaces listed are the logical interfaces, not the physical interfaces. For example, the output shows xe-0/0/0.0 instead of xe-0/0/0. This is because Junos OS creates VLANs on logical interfaces, not directly on physical interfaces.

Related Documentation

- Example: Setting Up Bridging with Multiple VLANs on page 102
- Understanding Bridging and VLANs on page 5

Example: Setting Up Bridging with Multiple VLANs

The QFX Series products use bridging and virtual LANs (VLANs) to connect network devices in a LAN—storage devices, file servers, and other network components—and to segment the LAN into smaller bridging domains.

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on a switch. Each VLAN is a collection of network nodes. When you use VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN, and only frames not destined for the local VLAN are forwarded to other broadcast domains. VLANs thus limit the amount of traffic flowing across the entire LAN, reducing the possible number of collisions and packet retransmissions within the LAN.

NOTE: This task uses Junos OS for QFX3500 and QFX3600 switches does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Example: Setting Up Bridging with Multiple VLANs*.

This example describes how to configure bridging for the QFX Series and how to create two VLANs to segment the LAN:

Requirements

This example uses the following hardware and software components:

- A configured and provisioned QFX3500 switch
- Junos OS Release 11.1 or later for the QFX Series

Overview and Topology

Switches connect all devices in an office or data center into a single LAN to provide sharing of common resources such as file servers. The default configuration creates a single VLAN, and all traffic on the switch is part of that broadcast domain. Creating separate network segments reduces the span of the broadcast domain and enables you to group related users and network resources without being limited by physical cabling or by the location of a network device in the building or on the LAN.

This example shows a simple configuration to illustrate the basic steps for creating two VLANs on a single switch. One VLAN, called **sales**, is for the sales and marketing group, and a second, called **support**, is for the customer support team. The sales and support groups each have their own dedicated file servers and other resources. For the switch ports to be segmented across the two VLANs, each VLAN must have its own broadcast domain, identified by a unique name and tag (VLAN ID). In addition, each VLAN must be on its own distinct IP subnet.

The topology used in this example consists of a single QFX3500 switch, with a total of 48 10-Gbps Ethernet ports. (For the purposes of this example, the QSFP+ ports Q0-Q3, which are ports xe-0/1/0 through xe-0/1/15, are excluded.)

Table 13: Components of the Multiple VLAN Topology

Property	Settings
Switch hardware	QFX3500 switch configured with 48 10-Gbps Ethernet ports (**xe-0/0/0** through **xe-0/0/47**)

Table 13: Components of the Multiple VLAN Topology *(continued)*

Property	Settings
VLAN names and tag IDs	**sales**, tag **100** **support**, tag **200**
VLAN subnets	**sales**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **support**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Interfaces in VLAN **sales**	File servers: **xe-0/0/20** and **xe-0/0/21**
Interfaces in VLAN **support**	File servers: **xe-0/0/46** and **xe-0/0/47**
Unused interfaces	**xe-0/0/2** and **xe-0/0/25**

This configuration example creates two IP subnets, one for the sales VLAN and the second for the support VLAN. The switch bridges traffic within a VLAN. For traffic passing between two VLANs, the switch routes the traffic using a Layer 3 routing interface on which you have configured the address of the IP subnet.

To keep the example simple, the configuration steps show only a few devices in each of the VLANs. Use the same configuration procedure to add more LAN devices.

Configuration

CLI Quick Configuration

To quickly configure Layer 2 switching for the two VLANs (**sales** and **support**) and to quickly configure Layer 3 routing of traffic between the two VLANs, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/3 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/22 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/20 unit 0 description “Sales file server port”
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members support
set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members support
set interfaces xe-0/0/44 unit 0 family ethernet-switching vlan members support
set interfaces xe-0/0/46 unit 0 description “Support file server port”
set interfaces xe-0/0/46 unit 0 family ethernet-switching vlan members support
set interfaces vlan unit 0 family inet address 192.0.2.0/25
set interfaces vlan unit 1 family inet address 192.0.2.128/25
set vlans sales l3–interface vlan.0
set vlans sales vlan-id 100
set vlans support vlan-id 200
set vlans support l3-interface vlan.1
```

Step-by-Step Procedure

Configure the switch interfaces and the VLANs to which they belong. By default, all interfaces are in access mode, so you do not have to configure the port mode.

1. Configure the interface for the file server in the **sales** VLAN:

```
[edit interfaces xe-0/0/20 unit 0]
user@switch# set description “Sales file server port”
user@switch# set family ethernet-switching vlan members sales
```

2. Configure the interface for the file server in the **support** VLAN:

```
[edit interfaces xe-0/0/46 unit 0]
user@switch# set description “Support file server port”
user@switch# set family ethernet-switching vlan members support
```

3. Create the subnet for the **sales** broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 0 family inet address 192.0.2.1/25
```

4. Create the subnet for the **support** broadcast domain:

```
[edit interfaces]
user@switch# set vlan unit 1 family inet address 192.0.2.129/25
```

5. Configure the VLAN tag IDs for the **sales** and **support** VLANs:

```
[edit vlans]
user@switch# set sales vlan-id 100
user@switch# set support vlan-id 200
```

6. To route traffic between the **sales** and **support** VLANs, define the interfaces that are members of each VLAN and associate a Layer 3 interface:

```
[edit vlans]
user@switch# set sales l3-interface vlan.0
user@switch# set support l3-interface vlan.1
```

Display the results of the configuration:

```
user@switch> show configuration
interfaces {
    xe-0/0/20 {
        unit 0 {
            description “Sales file server port”;
            family ethernet-switching {
                vlan members sales;
            }
        }
    }
    xe-0/0/46 {
        unit 0 {
            description “Support file server port”;
            family ethernet-switching {
                vlan members support;
            }
        }
        vlans {
            unit 0 {
                family inet address 192.0.2.1/25;
            }
            unit 1 {
                family inet address 192.0.2.129/25;
            }
```

```
    }
  }
}
vlans {
  sales {
    vlan-id 100;
    interface xe-0/0/0.0:
    interface xe-0/0/3/0;
    interface xe-0/0/20.0;
    interface xe-0/0/22.0;
    l3-interface vlan 0;
  }
  support {
    vlan-id 200;
    interface xe-0/0/24.0:
    interface xe-0/0/26.0;
    interface xe-0/0/44.0;
    interface xe-0/0/46.0;
    l3-interface vlan 1;
  }
}
```

TIP: To quickly configure the sales and support VLAN interfaces, issue the **load merge terminal** command. Then copy the hierarchy and paste it into the switch terminal window.

Verification

Verify that the **sales** and **support** VLANs have been created and are operating properly, perform these tasks:

- Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces on page 106
- Verifying That Traffic Is Being Routed Between the Two VLANs on page 107
- Verifying That Traffic Is Being Switched Between the Two VLANs on page 107

Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces

Purpose Verify that the **sales** and **support** VLANs have been created on the switch and that all connected interfaces on the switch are members of the correct VLAN.

Action To list all VLANs configured on the switch, use the **show vlans** command:

```
user@switch> show vlans
Name            Tag     Interfaces
default
                        xe-0/0/1.0, xe-0/0/2.0, xe-0/0/4.0, xe-0/0/5.0,
                        xe-0/0/6.0,  xe-0/0/7.0, xe-0/0/8.0, xe-0/0/9.0,
                        xe-0/0/10.0*, xe-0/0/11.0, xe-0/0/12.0, xe-0/0/13.0*,
                        xe-0/0/14.0, xe-0/0/15.0, xe-0/0/16.0, xe-0/0/17.0,
```

```
                        xe-0/0/18.0, xe-0/0/19.0, xe-0/0/21.0, xe-0/0/23.0*,
                        xe-0/0/25.0, xe-0/0/27.0, xe-0/0/28.0, xe-0/0/29.0,
                        xe-0/0/30.0, xe-0/0/31.0, xe-0/0/32.0, xe-0/0/33.0,
                        xe-0/0/34.0, xe-0/0/35.0, xe-0/0/36.0, xe-0/0/37.0,
                        xe-0/0/38.0, xe-0/0/39.0, xe-0/0/40.0, xe-0/0/41.0,
                        xe-0/0/42.0, xe-0/0/43.0, xe-0/0/45.0, xe-0/0/47.0,
                        xe-0/1/0.0*, xe-0/1/1.0*, xe-0/1/2.0*, xe-0/1/3.0*
sales       100
                        xe-0/0/0.0*, xe-0/0/3.0, xe-0/0/20.0, xe-0/0/22.0
support     200
                        xe-0/0/0.24, xe-0/0/26.0, xe-0/0/44.0, xe-0/0/46.0*
mgmt
                        me0.0*
```

Meaning The **show vlans** command lists all VLANs configured on the switch and which interfaces are members of each VLAN. This command output shows that the **sales** and **support** VLANs have been created. The **sales** VLAN has a tag ID of 100 and is associated with interfaces **xe-0/0/0.0**, **xe-0/0/3.0**, **xe-0/0/20.0**, and **xe-0/0/22.0**. VLAN **support** has a tag ID of 200 and is associated with interfaces **xe-0/0/24.0**, **xe-0/0/26.0**, **xe-0/0/44.0**, and **xe-0/0/46.0**.

Verifying That Traffic Is Being Routed Between the Two VLANs

Purpose Verify routing between the two VLANs.

Action List the Layer 3 routes in the switch Address Resolution Protocol (ARP) table:

```
user@switch> show arp
MAC Address       Address         Name                    Flags

00:00:0c:06:2c:0d       192.0.2.3       vlan.0                  None
00:13:e2:50:62:e0       192.0.2.11      vlan.1                  None
```

Meaning Sending IP packets on a multiaccess network requires mapping from an IP address to a MAC address (the physical or hardware address). The ARP table displays the mapping between the IP address and MAC address for both **vlan.0** (associated with **sales**) and **vlan.1** (associated with **support**). These VLANs can route traffic to each other.

Verifying That Traffic Is Being Switched Between the Two VLANs

Purpose Verify that learned entries are being added to the Ethernet switching table.

Action List the contents of the Ethernet switching table:

```
user@switch> show ethernet-switching table

Ethernet-switching table: 8 entries, 5 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:00:05:00:00:01 Learn          - xe-0/0/10.0
  default           00:00:5e:00:01:09 Learn          - xe-0/0/13.0
```

```
default          00:19:e2:50:63:e0 Learn          - xe-0/0/23.0
sales            *                 Flood          - All-members
sales            00:00:5e:00:07:09 Learn          - xe-0/0/0.0
support          *                 Flood          - All-members
support          00:00:5e:00:01:01 Learn          - xe-0/0/46.0
```

Meaning The output shows that learned entries for the **sales** and **support** VLANs have been added to the Ethernet switching table, and are associated with interfaces **xe-0/0/0.0** and **xe-0/0/46.0**. Even though the VLANs were associated with more than one interface in the configuration, these interfaces are the only ones that are currently operating.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- *Understanding Bridging*

CHAPTER 9

MAC Learning Configuration Examples

Example: Disabling MAC Learning

By default, MAC learning is enabled on the QFX Series. This topic provides examples for disabling, enabling, and verifying the operation of MAC learning on the QFX Series. These examples require that you be logged in as the root user to the switch on which you wish to modify MAC learning.

NOTE: This task uses Junos OS for QFX3500 and QFX3600 switches does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Example: Disabling MAC Learning*.

- To disable MAC learning in a VLAN:

```
[edit]
user@switch# edit ethernet-switching-options interfaces xe-0/0/0.0
[edit ethernet-switching-options interfaces xe-0/0/0.0]
user@switch# set no-mac-learning
```

- To reenable MAC learning:

```
[edit]
user@switch# edit ethernet-switching-options interfaces xe-0/0/0.0
[edit ethernet-switching-options interfaces xe-0/0/0.0]
user@switch# delete no-mac-learning
```

- To verify the status of MAC learning on the QFX Series:

```
user@switch> show ethernet-switching table
Learning stats: 10 learn msg rcvd, 2 error, 0 forced update
Interface              Local pkts        Transit pkts      Error
xe-0/0/0.0        0                6                 1
xe-0/0/22.0       0                0                 0
xe-0/0/1.0        0                4                 1
xe-0/0/2.0        0                0                 0
xe-0/0/3.0        0                0                 0
xe-0/0/4.0        0                0                 0
xe-0/0/19.0       0                0                 0
xe-0/0/18.0       0                0                 0
```

```
xe-0/0/9.0          0                   0                   0
```

Related Documentation

- Understanding MAC Learning on page 26
- Disabling MAC Learning on page 241
- no-mac-learning (Per VLAN) on page 286

Example: Disabling MAC Learning in a VLAN

When MAC learning is enabled, a MAC address is learned dynamically from a packet's source MAC address. By default, MAC learning is enabled on a VLAN. This topic provides examples for disabling, enabling, and verifying the operation of MAC learning in a VLAN. Disabling dynamic MAC learning in a VLAN on a QFX Series product prevents a node from learning source and destination MAC addresses. These examples require that you be logged in as the root user to the switch on which you wish to modify MAC learning. This example uses a VLAN named *blue*.

- To disable MAC learning in a VLAN:

  ```
  [edit vlans vlan-name]
  user@switch# set no-mac-learning
  ```

  For example:

  ```
  [edit vlans blue]
  user@switch# set no-mac-learning
  ```

- To verify that you have disabled MAC learning, issue the **show ethernet-switching table** command:

  ```
  user@switch> show ethernet-switching table
  Ethernet-switching table: 5 entries, 2 learned
    VLAN              MAC address       Type         Age Interfaces
    blue              *                 Flood          - All-members
    blue              00:1f:12:39:90:80 Static         - Router
    default           *                 Flood          - All-members
    default           00:1f:12:39:90:89 Learn       3:15 ge-0/0/1.0
    default           00:1f:12:39:a3:81 Learn          0 ge-0/0/1.0
  ```

  The CLI output shows that the VLAN named *blue* is not configured for MAC learning. The **Type** column includes only **static** (MAC address that are manually created) and **flood** (MAC addresses that are unknown and flooded to all members of the VLAN) entries.

- To reenable MAC learning in a VLAN, issue either of the following two commands::

  ```
  [edit vlans vlan-name]
  user@switch delete no-mac-learning
  user@switch# deactivate no-mac-learning
  ```

  For example:

  ```
  [edit vlans blue]
  user@switch delete no-mac-learning
  user@switch# deactivate no-mac-learning
  ```

- To verify that you have enabled MAC learning, issue the **show ethernet-switching table** command:

```
user@switch> show ethernet-switching table
Ethernet-switching table: 6 entries, 3 learned
  VLAN              MAC address       Type         Age Interfaces
  blue              *                 Flood          - All-members
  blue              00:1f:12:39:90:80 Static         - Router
  blue              00:1f:12:39:a3:80 Learn          0 ge-0/0/9.0
  default           *                 Flood          - All-members
  default           00:1f:12:39:90:89 Learn          0 ge-0/0/1.0
  default           00:1f:12:39:a3:81 Learn          0 ge-0/0/1.0
```

The CLI output shows that the VLAN named *blue* is configured for MAC learning. The **Type** column includes **static** (MAC address that are manually created), **flood** (MAC addresses that are unknown and flooded to all members of the VLAN), and .**Learn** (MAC addresses that are earned dynamically from a packet's source MAC address) entries.

Related Documentation

- Understanding MAC Learning on page 26
- Disabling MAC Learning in a VLAN on page 241
- no-mac-learning (Per VLAN) on page 286
- show ethernet-switching table on page 408

CHAPTER 10

MVRP Configuration Example

Example: Configuring Automatic VLAN Administration Using MVRP

As the numbers of servers and VLANs attached to a QFabric systems increase, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple redundant server Node group devices becomes increasingly difficult. To partially automate VLAN administration, you can enable Multiple VLAN Registration Protocol (MVRP) on your QFabric system. If your QFabric system connects to servers that host many virtual machines that require their own VLANs, using MVRP can save you the time and effort that would be required to manually configure and administer the VLANs on the interfaces that connect to the servers. For example, if a virtual machine moves between servers—and therefore connects to a different redundant server Node group interface—MVRP can configure the appropriate VLAN membership on the new server Node group interface.

NOTE: **Only trunk interfaces can be enabled for MVRP.**

This example describes how to configure MVRP on a QFabric system.

Requirements

This example uses the following hardware and software components:

- One QFabric system
- Junos OS Release 13.1 for the QFX Series

Overview and Topology

MVRP ensures that the VLAN membership information on the trunk interface is updated as the switch's access interfaces become active or inactive in the configured VLANs in a static or dynamic VLAN creation setup.

You do not need to explicitly bind a VLAN to the trunk interface. When MVRP is enabled, the trunk interface advertises all the VLANs that are active (bound to access interfaces) on that switch. An MVRP-enabled trunk interface does not advertise VLANs that have been configured on the switch but that are not currently bound to an access interface. Thus, MVRP provides the benefit of reducing network overhead—by limiting the scope of broadcast, unknown unicast, and multicast (BUM) traffic to interested devices only.

When VLAN access interfaces become active or inactive, MVRP ensures that the updated information is advertised on the trunk interface. Thus, in this example, distribution Switch C does not forward traffic to inactive VLANs.

A redundant server Node group device is connected to a server that hosts virtual machines for three customers, each of which requires its own VLAN.

- **customer-1**: VLAN ID 100
- **customer-2**: VLAN ID 200
- **customer-3**: VLAN ID 300

Table 10 on page 72 explains the components of the example topology.

Table 14: Components of the Example Topology

Settings	Settings
Hardware	• Redundant server Node group device • Network Node group device
VLAN names and IDs	• **customer-1**, VLAN ID (tag)**100** • **customer-2**, VLAN ID (tag)**200** • **customer-3**, VLAN ID (tag)**300**
Interfaces	Redundant server Node group device interfaces: • **RSNG:xe-0/1/1**—Uplink to interconnect device • **RSNG:xe-0/0/1**—Server-facing interface Network Node group device interface: • **NNG:xe-0/0/1**—Uplink to interconnect device

Configuring VLANs and Network Node Group Interfaces

To configure VLANs, bind the VLANs to the server-facing trunk interface, and enable MVRP on the trunk interface of the network Node group device, perform these tasks:

CLI Quick Configuration

To quickly configure VLANs on the QFabric system, assign VLAN membership to the uplink port on the network Node group device, and configure the uplink port to be trunk:

```
[edit]
set vlans customer-1 vlan-id 100
set vlans customer-2 vlan-id 200
set vlans customer-3 vlan-id 300
set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching vlan members [customer-1
customer-2 customer-3]
```

NOTE: As recommended as a best practice, default MVRP timers are used in this example, so they are not configured. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms for the leaveall timer. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Step-by-Step Procedure

To create the VLANs and configure the network Node group device for MVRP, follow these steps. Note that you are creating VLANs for the entire QFabric system, so you do not need to create them on specific QFabric devices.

1. Configure the VLAN for customer 1:

   ```
   [edit]
   user@qfabric# set vlans customer-1 vlan-id 100
   ```

2. Configure the VLAN for customer 2:

   ```
   [edit]
   user@qfabric# set vlans customer-2 vlan–id 200
   ```

3. Configure the VLAN for customer 3:

   ```
   [edit]
   user@qfabric# set vlans customer-3 vlan–id 300
   ```

4. Configure an uplink interface (one that connects to an interconnect device) to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/0/1 unit 0 family ethernet-switching port-mode
   trunk
   ```

5. Configure the uplink interface to be a member of all three VLANs:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/0/1 unit 1 family ethernet-switching vlan members
   [customer-1 customer-2 customer-3]
   ```

NOTE: If you want the uplink interface to be a member of all the VLANs in the QFabric system, you can enter all instead of specifying the individual VLANs.

Results

Check the results of the configuration on the network Node group device:

```
[edit]
user@qfabric# show interfaces NNG:xe-0/0/1.0
family ethernet-switching {
   port-mode trunk;
   vlan {
      members customer-1 customer-2 customer-3;
   }
}

[edit]
user@qfabric# show vlans
customer-1 {
   vlan-id 100;
}
customer-2 {
   vlan-id 200;
}
customer-3 {
   vlan-id 300;
}
```

Configuring the Redundant Server Node Group

CLI Quick Configuration

To quickly configure the redundant server Node group device for MVRP:

```
[edit]
set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching vlan members [customer-1
customer-2 customer-3]
set interfaces RSNG:xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set protocols mvrp interface RSNG:xe-0/0/1.0 passive
```

Step-by-Step Procedure

To configure the redundant server Node group device, follow these steps. Note that you do not need to configure the VLANs on the server-facing interface (RSNG:xe-0/0/1), but you do need to configure the VLANs on the uplink interface. Also notice that in this example you configure the server-facing interface to be passive, which means that it will not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from the server.

1. Configure an uplink interface (one that connects to the interconnect device) to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces RSNG:xe-0/1/1 unit 0 family ethernet-switching port-mode
   trunk
   ```

2. Configure the uplink interface to be a member of all three VLANs:

   ```
   [edit]
   user@qfabric# set interfaces NNG:xe-0/1/1 unit 0 family ethernet-switching vlan members
   [customer-1 customer-2 customer-3]
   ```

3. Configure an interface that connects to the server that hosts multiple virtual machines to be a trunk:

   ```
   [edit]
   user@qfabric# set interfaces RSNG:xe-0/0/1 unit 0 family ethernet-switching port-mode
   trunk
   ```

4. Enable MVRP on the server-facing trunk interface and configure it to be passive:

```
[edit]
user@qfabric# set protocols mvrp interface RSNG:xe-0/0/1.0 passive
```

Results Check the results of the configuration for the redundant server Node group:

```
[edit]
user@qfabric# show interfaces RSNG:xe-0/0/1.0
family ethernet-switching {
   port-mode trunk;
}

[edit]
user@qfabric# show interfaces RSNG:xe-0/1/1.0
family ethernet-switching {
   port-mode trunk;
   }
passive
}

[edit]
user@qfabric# show protocols mvrp
interface RSNG:xe-0/0/1.0;
```

Verification

To confirm that the configuration is updating VLAN membership, perform these tasks:

- Verifying That MVRP Is Enabled On The QFabric System on page 117

Verifying That MVRP Is Enabled On The QFabric System

Purpose Verify that MVRP is enabled on the appropriate interfaces

Action Show the MVRP configuration:

```
user@qfabric> show mvrp

MVRP configuration
MVRP status                 : Enabled

MVRP timers (ms):
Interface         Join   Leave       LeaveAll
--------------    -----  --------    -----------
NNG:xe-0/0/1.0    200     1000       10000
RSNG:xe-0/0/1.0   200     1000       10000
RSNG:xe-0/1/1.0   200     1000       10000


Interface           Status        Registration Mode
--------------  --------      -----------------
NNG:xe-0/0/1.0      Enabled       Normal
RSNG:xe-0/1/1.0     Enabled       Normal
RSNG:xe-0/0/1.0     Enabled       Passive
```

Meaning The results show that MVRP is enabled on the appropriate network Node group and redundant server Node group interfaces and that the default timers are used.

Related Documentation

- Understanding Multiple VLAN Registration Protocol (MVRP) on page 47

CHAPTER 11

Private VLAN Configuration Examples

Example: Configuring a Private VLAN on a Single Switch

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and even to limit the communication between known hosts. The private VLAN (PVLAN) feature allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN.

This example describes how to create a PVLAN on a single switch:

Requirements

This example uses the following hardware and software components:

- One QFX3500 device
- Junos OS Release 12.1 or later for the QFX Series

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs" on page 220.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows a simple topology to illustrate how to create a PVLAN with one primary VLAN and two community VLANs, one for HR and one for finance, as well as two isolated ports—one for the mail server and the other for the backup server.

Table 15 on page 120 lists the settings for the sample topology.

Table 15: Components of the Topology for Configuring a PVLAN

Interface	Description
ge-0/0/0.0	Primary VLAN (**pvlan100**) trunk interface
ge-0/0/11.0	User 1, HR Community (**hr-comm**)
ge-0/0/12.0	User 2, HR Community (**hr-comm**)
ge-0/0/13.0	User 3, Finance Community (**finance-comm**)
ge-0/0/14.0	User 4, Finance Community (**finance-comm**)
ge-0/0/15.0	Mail server, Isolated (**isolated**)
ge-0/0/16.0	Backup server, Isolated (**isolated**)
ge-1/0/0.0	Primary VLAN (**pvlan100**) trunk interface

Configuration

CLI Quick Configuration

To quickly create and configure a PVLAN, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans pvlan100 vlan-id 100
set interfaces ge-0/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members pvlan
set interfaces ge-1/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members pvlan
set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/12 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/13 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/14 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/15 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/16 unit 0 family ethernet-switching port-mode access
set vlans pvlan100 pvlan
set vlans pvlan100 interface ge-0/0/0.0
set vlans pvlan100 interface ge-1/0/0.0
set vlans hr-comm interface ge-0/0/11.0
set vlans hr-comm interface ge-0/0/12.0
set vlans finance-comm interface ge-0/0/13.0
set vlans finance-comm interface ge-0/0/14.0
set vlans hr-comm primary-vlan pvlan100
set vlans finance-comm primary-vlan pvlan100
set pvlan100 interface ge-0/0/15.0 isolated
set pvlan100 interface ge-0/0/16.0 isolated
```

Step-by-Step Procedure

To configure the PVLAN:

1. Set the VLAN ID for the primary VLAN:

```
[edit vlans]
user@switch# set pvlan vlan-id 100
```

2. Set the interfaces and port modes:

```
[edit interfaces]
user@switch# set ge-0/0/0 unit 0 family ethernet-switching port-mode trunk
user@switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members pvlan
user@switch# set ge-1/0/0 unit 0 family ethernet-switching port-mode trunk
user@switch# set ge-1/0/0 unit 0 family ethernet-switching vlan members pvlan
user@switch# set ge-0/0/11 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/12 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/13 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/14 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/15 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/16 unit 0 family ethernet-switching port-mode access
```

3. Set the primary VLAN to have no local switching:

NOTE: The primary VLAN must be a tagged VLAN.

```
[edit vlans]
user@switch# set pvlan100 pvlan
```

4. Add the trunk interfaces to the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/0.0
user@switch# set pvlan100 interface ge-1/0/0.0
```

5. For each secondary VLAN, configure access interfaces:

NOTE: We recommend that the secondary VLANs be untagged VLANs. It does not impair functioning if you tag the secondary VLANS. However, the tags are not used when a secondary VLAN is configured on a single switch.

```
[edit vlans]
user@switch# set hr-comm interface ge-0/0/11.0
user@switch# set hr-comm interface ge-0/0/12.0
user@switch# set finance-comm interface ge-0/0/13.0
user@switch# set finance-comm interface ge-0/0/14.0
```

6. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set hr-comm primary-vlan pvlan100
user@switch# set finance-comm primary-vlan pvlan100
```

7. Configure the isolated interfaces in the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/15.0 isolated
user@switch# set pvlan100 interface ge-0/0/16.0 isolated
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                }
            }
        }
    }
    ge-1/0/0 {
        unit 0 {
            family ethernet-switching;
        }
    }
    ge-0/0/11 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    ge-0/0/12 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    ge-0/0/13 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    ge-0/0/14 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
vlans {
    finance-comm {
    interface {
        ge-0/0/13.0;
        ge-0/0/14.0;
```

```
    }
    primary-vlan pvlan100;
    }
    hr-comm {
      interface {
        ge-0/0/11.0;
        ge-0/0/12.0;
      }
      primary-vlan pvlan100;
    }
    pvlan100 {
      vlan-id 100;
      interface {
        ge-0/0/15.0;
        ge-0/0/16.0;
        ge-0/0/0.0;
        ge-1/0/0.0;
      }
      pvlan;
    }
  }
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That the Private VLAN and Secondary VLANs Were Created on page 123

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose Verify that the primary VLAN and secondary VLANs were properly created on the switch.

Action Use the **show vlans** command:

```
user@switch> show vlans pvlan100 extensive
VLAN: pvlan100, Created at: Tue Sep 16 17:59:47 2008
802.1Q Tag: 100, Internal index: 18, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  6 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/11.0, untagged, access
      ge-0/0/12.0, untagged, access
      ge-0/0/13.0, untagged, access
      ge-0/0/14.0, untagged, access
      ge-0/0/15.0, untagged, access
      ge-0/0/16.0, untagged, access
      ge-1/0/0.0, tagged, trunk
Secondary VLANs: Isolated 2, Community  2
  Isolated VLANs :
      __pvlan_pvlan_ge-0/0/15.0__
      __pvlan_pvlan_ge-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm

user@switch> show vlans hr-comm extensive
```

```
VLAN: hr-comm, Created at: Tue Sep 16 17:59:47 2008
Internal index: 22, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/11.0, untagged, access
      ge-0/0/12.0, untagged, access
      ge-1/0/0.0, tagged, trunk

user@switch> show vlans finance-comm extensive
VLAN: finance-comm, Created at: Tue Sep 16 17:59:47 2008
Internal index: 21, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/13.0, untagged, access
      ge-0/0/14.0, untagged, access
      ge-1/0/0.0, tagged, trunk

user@switch> show vlans __pvlan_pvlan_ge-0/0/15.0__ extensive
VLAN: __pvlan_pvlan_ge-0/0/15.0__, Created at: Tue Sep 16 17:59:47 2008
Internal index: 19, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  1 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/15.0, untagged, access
      ge-1/0/0.0, tagged, trunk

user@switch> show vlans __pvlan_pvlan_ge-0/0/16.0__ extensive
VLAN: __pvlan_pvlan_ge-0/0/16.0__, Created at: Tue Sep 16 17:59:47 2008
Internal index: 20, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  1 (Active = 0)
      ge-0/0/0.0, tagged, trunk
      ge-0/0/16.0, untagged, access
      ge-1/0/0.0, tagged, trunk
```

Meaning The output shows that the primary VLAN was created and identifies the interfaces and secondary VLANs associated with it.

Related Documentation

- Understanding Private VLANs on page 28
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 33
- Creating a Private VLAN on a Single Switch on page 247

Example: Configuring a Private VLAN Spanning Multiple Switches

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and even to limit the communication between known hosts. The private VLAN (PVLAN) feature allows an administrator to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN. A PVLAN can span multiple switches.

This example describes how to create a PVLAN spanning multiple switches. The example creates one primary PVLAN, containing multiple secondary VLANs:

Requirements

This example uses the following hardware and software components:

- Three QFX3500 devices
- Junos OS Release 12.1 or later for the QFX Series

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs" on page 220.

Overview and Topology

In a large office with multiple buildings and VLANs, you might need to isolate some workgroups or other endpoints for security reasons or to partition the broadcast domain. This configuration example shows how to create a PVLAN spanning multiple QFX devices, with one primary VLAN containing two community VLANs (one for HR and one for Finance), and an interswitch isolated VLAN (for the mail server, the backup server, and the CVS server). The PVLAN comprises three switches, two access switches and one distribution switch. The PVLAN is connected to a router through a promiscuous port, which is configured on the distribution switch.

NOTE: The isolated ports on Switch 1 and on Switch 2 do not have Layer 2 connectivity with one another even though they are included within the same domain. See "Understanding Private VLANs" on page 28.

Figure 11 on page 126 shows the topology for this example—two access switches connecting to a distribution switch, which has a connection (through a promiscuous port) to the router.

Figure 11: PVLAN Topology Spanning Multiple Switches

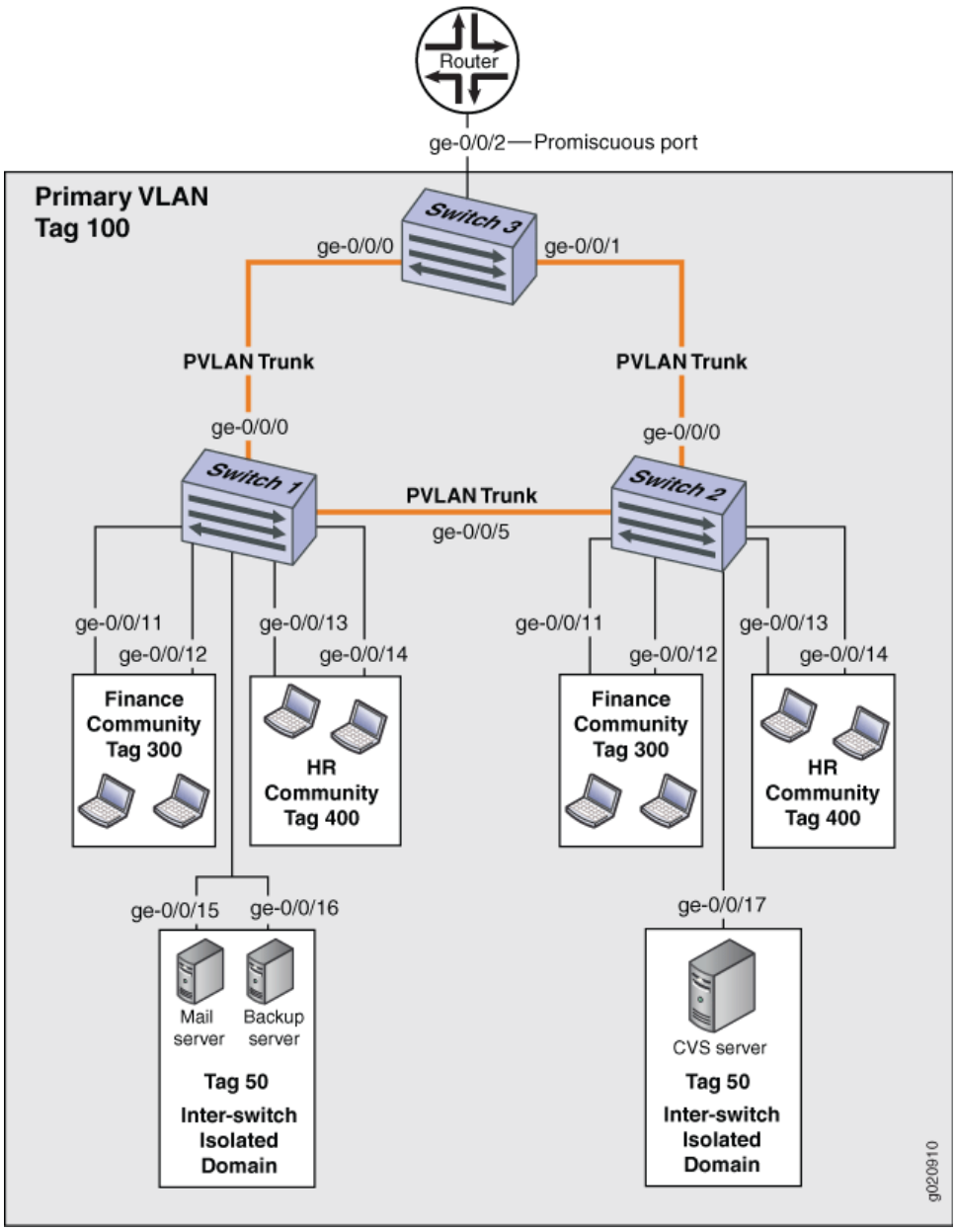


Table 16 on page 127, Table 17 on page 127, and Table 18 on page 128 list the settings for the example topology.

Table 16: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, connects Switch 1 to Switch 3 **ge-0/0/5.0**, connects Switch 1 to Switch 2
Isolated Interfaces in primary VLAN	**ge-0/0/15.0**, mail server **ge-0/0/16.0**, backup server
Interfaces in VLAN **finance-com**	**ge-0/0/11.0** **ge-0/0/12.0**
Interfaces in VLAN **hr-comm**	**ge-0/0/13.0** **ge-0/0/14.0**

Table 17: Components of Switch 2 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, connects Switch 2 to Switch 3 **ge-0/0/5.0**, connects Switch 2 to Switch 1
Isolated Interface in primary VLAN	**ge-0/0/17.0**, CVS server
Interfaces in VLAN **finance-com**	**ge-0/0/11.0** **ge-0/0/12.0**
Interfaces in VLAN **hr-comm**	**ge-0/0/13.0** **ge-0/0/14.0**

Table 18: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple Devices

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
PVLAN trunk interfaces	**ge-0/0/0.0**, connects Switch 3 to Switch 1 **ge-0/0/1.0**, connects Switch 3 to Switch 2
Promiscuous port	**ge-0/0/2**, connects the PVLAN to the router NOTE: You must configure the trunk port that connects the PVLAN to another switch or router outside the PVLAN as a member of the PVLAN, which implicitly configures it as a promiscuous port.

Configuring a PVLAN on Switch 1

When configuring a PVLAN on multiple switches, these rules apply:

- The primary VLAN must be a tagged VLAN. We recommend that you configure the primary VLAN first.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port. You must also configure the primary VLAN to be private using the pvlan statement.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.

CLI Quick Configuration

To quickly create and configure a PVLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 1:

```
[edit]
set vlans finance-comm vlan-id 300
set vlans finance-comm interface ge-0/0/11.0
set vlans finance-comm interface ge-0/0/12.0
set vlans finance-comm primary-vlan pvlan100
set vlans hr-comm vlan-id 400
set vlans hr-comm interface ge-0/0/13.0
set vlans hr-comm interface ge-0/0/14.0
set vlans hr-comm primary-vlan pvlan100
set vlans pvlan100 vlan-id 100
set vlans pvlan100 interface ge-0/0/15.0
set vlans pvlan100 interface ge-0/0/16.0
set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk
set vlans pvlan100 interface ge-0/0/5.0 pvlan-trunk
set vlans pvlan100 pvlan
set vlans pvlan100 pvlan isolation-vlan-id 50
set pvlan100 interface ge-0/0/15.0 isolated
set pvlan100 interface ge-0/0/16.0 isolated
```

Step-by-Step Procedure

1. Set the VLAN ID for the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 vlan-id 100
```

2. Set the PVLAN trunk interfaces to connect this VLAN across neighboring switches:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/0.0 pvlan-trunk
user@switch# set pvlan100 interface ge-0/0/5.0 pvlan-trunk
```

3. Set the primary VLAN to be private and have no local switching:

```
[edit vlans]
user@switch# set pvlan100 pvlan
```

4. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

```
[edit vlans]
user@switch# set finance-comm vlan-id 300
```

5. Configure access interfaces for the **finance-comm** VLAN:

```
[edit vlans]
user@switch# set finance-comm interface ge-0/0/11.0
user@switch# set finance-comm interface ge-0/0/12.0
```

6. Set the primary VLAN of this secondary community VLAN, **finance-comm** :

```
[edit vlans]
user@switch# set vlans finance-comm primary-vlan pvlan100
```

7. Set the VLAN ID for the HR community VLAN that spans the switches.

```
[edit vlans]
user@switch# set hr-comm vlan-id 400
```

8. Configure access interfaces for the **hr-comm** VLAN:

```
[edit vlans]
user@switch# set hr-comm interface ge-0/0/13.0
user@switch# set hr-comm interface ge-0/0/14.0
```

9. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

```
[edit vlans]
user@switch# set vlans hr-comm primary-vlan pvlan100
```

10. Set the interswitch isolated ID to create an interswitch isolated domain that spans the switches:

```
[edit vlans]
user@switch# set pvlan100 pvlan isolation-vlan-id 50
```

11. Configure the isolated interfaces in the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/15.0 isolated
user@switch# set pvlan100 interface ge-0/0/16.0 isolated
```

NOTE: When you configure an isolated port, include it as a member of the primary VLAN, but do not configure it as a member of any community VLAN.

Results

Check the results of the configuration:

```
[edit]
  user@switch# show
vlans {
  finance-comm {
    vlan-id 300;
    interface {
      ge-0/0/11.0;
      ge-0/0/12.0;
    }
    primary-vlan pvlan100;
  }
  hr-comm {
    vlan-id 400;
    interface {
      ge-0/0/13.0;
      ge-0/0/14.0;
    }
    primary-vlan pvlan100;
  }
  pvlan100 {
    vlan-id 100;
    interface {
      ge-0/0/15.0;
      ge-0/0/16.0;
      ge-0/0/0.0 {
        pvlan-trunk;
      }
      ge-0/0/5.0 {
        pvlan-trunk;
      }
    }
    pvlan;
    isolation-vlan-id 50;
  }
}
```

Configuring a PVLAN on Switch 2

CLI Quick Configuration

To quickly create and configure a private VLAN spanning multiple switches, copy the following commands and paste them into the terminal window of Switch 2:

NOTE: The configuration of Switch 2 is the same as the configuration of Switch 1 except for the interface in the interswitch isolated domain. For Switch 2, the interface is ge-0/0/17.0.

```
[edit]
set vlans finance-comm vlan-id 300
set vlans finance-comm interface ge-0/0/11.0
set vlans finance-comm interface ge-0/0/12.0
set vlans finance-comm primary-vlan pvlan100
set vlans hr-comm vlan-id 400
set vlans hr-comm interface ge-0/0/13.0
```

```
set vlans hr-comm interface ge-0/0/14.0
set vlans hr-comm primary-vlan pvlan100
set vlans pvlan100 vlan-id 100
set vlans pvlan100 interface ge-0/0/17.0
set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk
set vlans pvlan100 interface ge-0/0/5.0 pvlan-trunk
set vlans pvlan100 pvlan
set vlans pvlan100 pvlan isolation-vlan-id 50
set pvlan100 interface ge-0/0/17.0 isolated
```

Step-by-Step Procedure

To configure a PVLAN on Switch 2 that will span multiple switches:

1. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   ```
   [edit vlans]
   user@switch# set finance-comm vlan-id 300
   ```

2. Configure access interfaces for the **finance-comm** VLAN:

   ```
   [edit vlans]
   user@switch# set finance-comm interface ge-0/0/11.0
   user@switch# set finance-comm interface ge-0/0/12.0
   ```

3. Set the primary VLAN of this secondary community VLAN, **finance-comm**:

   ```
   [edit vlans]
   user@switch# set vlans finance-comm primary-vlan pvlan100
   ```

4. Set the VLAN ID for the HR community VLAN that spans the switches.

   ```
   [edit vlans]
   user@switch# set hr-comm vlan-id 400
   ```

5. Configure access interfaces for the **hr-comm** VLAN:

   ```
   [edit vlans]
   user@switch# set hr-comm interface ge-0/0/13.0
   user@switch# set hr-comm interface ge-0/0/14.0
   ```

6. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

   ```
   [edit vlans]
   user@switch# set vlans hr-comm primary-vlan pvlan100
   ```

7. Set the VLAN ID for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set pvlan100 vlan-id 100
   ```

8. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

   ```
   [edit vlans]
   user@switch# set pvlan100 interface ge-0/0/0.0 pvlan-trunk
   user@switch# set pvlan100 interface ge-0/0/5.0 pvlan-trunk
   ```

9. Set the primary VLAN to be private and have no local switching:

   ```
   [edit vlans]
   user@switch# set pvlan100 pvlan
   ```

10. Set the interswitch isolated ID to create an interswitch isolated domain that spans the switches:

    ```
    [edit vlans]
    user@switch# set pvlan100 pvlan isolation-vlan-id 50
    ```

NOTE: To configure an isolated port, include it as one of the members of the primary VLAN, but do not configure it as belonging to one of the community VLANs.

11. Configure the isolated interface in the primary VLAN:

```
[edit vlans]
user@switch# set pvlan100 interface ge-0/0/17.0 isolated
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
    finance-comm {
        vlan-id 300;
        interface {
            ge-0/0/11.0;
            ge-0/0/12.0;
        }
        primary-vlan pvlan100;
    }
    hr-comm {
        vlan-id 400;
        interface {
            ge-0/0/13.0;
            ge-0/0/14.0;
        }
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/15.0;
            ge-0/0/16.0;
            ge-0/0/0.0 {
                pvlan-trunk;
            }
            ge-0/0/5.0 {
                pvlan-trunk;
            }
            ge-0/0/17.0;
        }
        pvlan;
        isolation-vlan-id 50;
    }
}
```

Configuring a PVLAN on Switch 3

CLI Quick Configuration

To quickly configure Switch 3 to function as the distribution switch of this PVLAN, copy the following commands and paste them into the terminal window of Switch 3:

NOTE: Interface ge-0/0/2.0 is a trunk port connecting the PVLAN to a router.

```
[edit]
set vlans finance-comm vlan-id 300
set vlans finance-comm primary-vlan pvlan100
set vlans hr-comm vlan-id 400
set vlans hr-comm primary-vlan pvlan100
set vlans pvlan100 vlan-id 100
set vlans pvlan100 interface ge-0/0/0.0 pvlan-trunk
set vlans pvlan100 interface ge-0/0/1.0 pvlan-trunk
set vlans pvlan100 pvlan
set vlans pvlan100 pvlan isolation-vlan-id 50
```

Step-by-Step Procedure

To configure Switch 3 to function as the distribution switch for this PVLAN, use the following procedure:

1. Set the VLAN ID for the **finance-comm** community VLAN that spans the switches:

   ```
   [edit vlans]
   user@switch# finance-comm vlan-id 300
   ```

2. Set the primary VLAN of this secondary community VLAN, **finance-comm**:

   ```
   [edit vlans]
   user@switch# set vlans finance-comm primary-vlan pvlan100
   ```

3. Set the VLAN ID for the HR community VLAN that spans the switches:

   ```
   [edit vlans]
   user@switch# set hr-comm vlan-id 400
   ```

4. Set the primary VLAN of this secondary community VLAN, **hr-comm**:

   ```
   [edit vlans]
   user@switch# set vlans hr-comm primary-vlan pvlan100
   ```

5. Set the VLAN ID for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set pvlan100 vlan-id 100
   ```

6. Set the PVLAN trunk interfaces that will connect this VLAN across neighboring switches:

   ```
   [edit vlans]
   user@switch# set pvlan100 interface ge-0/0/0.0 pvlan-trunk
   user@switch# set pvlan100 interface ge-0/0/5.0 pvlan-trunk
   ```

7. Set the primary VLAN to be private and have no local switching:

   ```
   [edit vlans]
   user@switch# set pvlan100 pvlan
   ```

8. Set the interswitch isolated ID to create an interswitch isolated domain that spans the switches:

   ```
   [edit vlans]
   user@switch# set pvlan100 pvlan isolation-vlan-id 50
   ```

NOTE: To configure an isolated port, include it as one of the members of the primary VLAN, but do not configure it as belonging to one of the community VLANs.

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
    finance-comm {
        vlan-id 300;
        primary-vlan pvlan100;
    }
    hr-comm {
        vlan-id 400;
        primary-vlan pvlan100;
    }
    pvlan100 {
        vlan-id 100;
        interface {
            ge-0/0/0.0 {
                pvlan-trunk;
            }
            ge-0/0/1.0 {
                pvlan-trunk;
            }
            ge-0/0/2.0;
        }
        pvlan;
        isolation-vlan-id 50;
    }
}
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 1 on page 135
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2 on page 136
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3 on page 137

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 1

Purpose Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 1:

Action Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
VLAN: __pvlan_pvlan100_ge-0/0/15.0__, Created at: Thu Sep 16 23:15:27 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/15.0*, untagged, access

VLAN: __pvlan_pvlan100_ge-0/0/16.0__, Created at: Thu Sep 16 23:15:27 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/16.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 300, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 400, Internal index: 9, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
```

```
VLAN: pvlan100, Created at: Thu Sep 16 23:15:27 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  6 (Active = 6)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
      ge-0/0/15.0*, untagged, access
      ge-0/0/16.0*, untagged, access
Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_ge-0/0/15.0__
      __pvlan_pvlan100_ge-0/0/16.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning The output shows that a PVLAN was created on Switch 1 and shows that it includes two isolated VLANs, two community VLANs, and an interswitch isolated VLAN. The presence of the pvlan-trunk and Inter-switch-isolated fields indicates that this PVLAN is spanning more than one switch.

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2

Purpose Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 2:

Action Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
VLAN: __pvlan_pvlan100_ge-0/0/17.0__, Created at: Thu Sep 16 23:19:22 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  1 (Active = 1)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/17.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 50, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
```

```
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 300, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 400, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  2 (Active = 2)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Thu Sep 16 23:19:22 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  5 (Active = 5)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/5.0*, tagged, trunk, pvlan-trunk
      ge-0/0/11.0*, untagged, access
      ge-0/0/12.0*, untagged, access
      ge-0/0/13.0*, untagged, access
      ge-0/0/14.0*, untagged, access
      ge-0/0/17.0*, untagged, access
Secondary VLANs: Isolated 1, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_ge-0/0/17.0__
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning The output shows that a PVLAN was created on Switch 2 and shows that it includes one isolated VLAN, two community VLANs, and an interswitch isolated VLAN. The presence of the pvlan-trunk and Inter-switch-isolated fields indicates that this PVLAN is spanning more than one switch. When you compare this output to the output of Switch 1, you can see that both switches belong to the same PVLAN (**pvlan100**).

Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3

Purpose Verify that the PVLAN configuration spanning multiple switches is working properly on Switch 3:

Action Use the **show vlans extensive** command:

```
user@switch> show vlans extensive
```

```
VLAN: __pvlan_pvlan100_isiv__, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 50, Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: default, Created at: Thu Sep 16 03:03:18 2010
Internal index: 2, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)

VLAN: finance-comm, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 300, Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: hr-comm, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 400, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: pvlan100
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk

VLAN: pvlan100, Created at: Thu Sep 16 23:22:40 2010
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 2 (Active = 2), Untagged  0 (Active = 0)
      ge-0/0/0.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, tagged, trunk, pvlan-trunk
Secondary VLANs: Isolated 0, Community  2, Inter-switch-isolated  1
  Community VLANs :
      finance-comm
      hr-comm
  Inter-switch-isolated VLAN :
      __pvlan_pvlan100_isiv__
```

Meaning The output shows that the PVLAN (**pvlan100**) is configured on Switch 3 and that it includes no isolated VLANs, two community VLANs, and an interswitch isolated VLAN. But Switch 3 is functioning as a distribution switch, so the output does not include access interfaces within the PVLAN. It shows only the **pvlan-trunk** interfaces that connect **pvlan100** from Switch 3 to the other switches (Switch 1 and Switch 2) in the same PVLAN.

Related Documentation

- Understanding Private VLANs on page 28
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 33
- Example: Configuring a Private VLAN on a Single Switch on page 119

Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports

This example shows how to configure secondary VLAN trunk ports and promiscuous access ports as part of a private VLAN configuration. Secondary VLAN trunk ports carry secondary VLAN traffic.

For a given private VLAN, a secondary VLAN trunk port can carry traffic for only one secondary VLAN. However, a secondary VLAN trunk port can carry traffic for multiple secondary VLANs as long as each secondary VLAN is a member of a different private (primary) VLAN. For example, a secondary VLAN trunk port can carry traffic for a community VLAN that is part of primary VLAN pvlan100 and also carry traffic for an isolated VLAN that is part of primary VLAN pvlan400.

To configure a trunk port to carry secondary VLAN traffic, use the isolated and **interface** statements, as shown in steps 12 and 13 of the example configuration for Switch 1.

NOTE: **When traffic egresses from a secondary VLAN trunk port, it normally carries the tag of the primary VLAN that the secondary port is a member of. If you want traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag, use the extend-secondary-vlan-id statement.**

A promiscuous access port carries untagged traffic and can be a member of only one primary VLAN. Traffic that ingresses on a promiscuous access port is forwarded to the ports of the secondary VLANs that are members of the primary VLAN that the promiscuous access port is a member of. This traffic carries the appropriate secondary VLAN tags when it egresses from the secondary VLAN ports if the secondary VLAN port is a trunk port.

To configure an access port to be promiscuous, use the promiscuous statement, as shown in step Figure 10 on page 45 of the example configuration for Switch 2.

If traffic ingresses on a secondary VLAN port and egresses on a promiscuous access port, the traffic is untagged on egress. If tagged traffic ingresses on a promiscuous access port, the traffic is discarded.

Requirements

This example uses the following hardware and software components:

- Two QFX devices

- Junos OS Release 12.2 or later for the QFX Series

Overview and Topology

Figure 12 on page 140 shows the topology used in this example. Switch 1 includes several primary and secondary private VLANs and also includes two secondary VLAN trunk ports configured to carry secondary VLANs that are members of primary VLANs pvlan100 and pvlan400.

Switch 2 includes the same private VLANs. The figure shows xe-0/0/0 on Switch 2 as configured with promiscuous access ports or promiscuous trunk ports. The example configuration included here configures this port as a promiscuous access port.

The figure also shows how traffic would flow after ingressing on the secondary VLAN trunk ports on Switch 1.

Figure 12: PVLAN Topology with Secondary VLAN Trunk Ports and Promiscuous Access Port

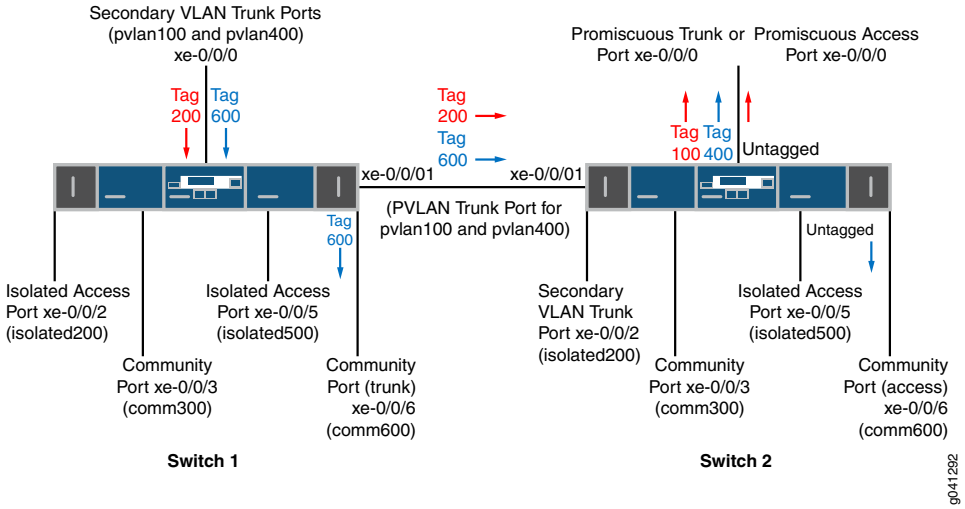


Table 19 on page 140 and Table 20 on page 141 list the settings for the example topology on both switches.

Table 19: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 1

Component	Description
pvlan100, ID 100	Primary VLAN
pvlan400, ID 400	Primary VLAN
comm300, ID 300	Community VLAN, member of pvlan100
comm600, ID 600	Community VLAN, member of pvlan400
isolation-vlan-id 200	VLAN ID for isolated VLAN, member of pvlan100

Table 19: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 1 *(continued)*

Component	Description
isolation–vlan-id 500	VLAN ID for isolated VLAN, member of pvlan400
xe-0/0/0.0	Secondary VLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/1.0	PVLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/2.0	Isolated access port for pvlan100
xe-0/0/3.0	Community access port for comm300
xe-0/0/5.0	Isolated access port for pvlan400
xe-0/0/6.0	Community trunk port for comm600

Table 20: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 2

Component	Description
pvlan100, ID 100	Primary VLAN
pvlan400, ID 400	Primary VLAN
comm300, ID 300	Community VLAN, member of pvlan100
comm600, ID 600	Community VLAN, member of pvlan400
isolation-vlan-id 200	VLAN ID for isolated VLAN, member of pvlan100
isolation–vlan-id 500	VLAN ID for isolated VLAN, member of pvlan400
xe-0/0/0.0	Promiscuous access port for primary VLANs pvlan100
xe-0/0/1.0	PVLAN trunk port for primary VLANs pvlan100 and pvlan400
xe-0/0/2.0	Secondary trunk port for isolated VLAN, member of pvlan100
xe-0/0/3.0	Community access port for comm300
xe-0/0/5.0	Isolated access port for pvlan400
xe-0/0/6.0	Community access port for comm600

Configuring the PVLANs on Switch 1

CLI Quick Configuration

To quickly create and configure the PVLANs on Switch 1, copy the following commands and paste them into a switch terminal window:

```
[edit]
```

```
set interfaces xe-0/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
set interfacesxe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/3 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/5 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode trunk
set vlans pvlan100 vlan-id 100
set vlans pvlan400 vlan-id 400
set vlans pvlan100 pvlan
set vlans pvlan400 pvlan
set vlans pvlan100 interface xe-0/0/1.0 pvlan-trunk
set vlans pvlan400 interface xe-0/0/1.0 pvlan-trunk
set vlans comm300 vlan-id 300
set vlans comm300 primary-vlan pvlan100
set vlans comm300 interface xe-0/0/3.0
set vlans comm600 vlan-id 600
set vlans comm600 primary-vlan pvlan400
set vlans comm600 interface xe-0/0/6.0
set vlans pvlan100 pvlan isolation-vlan-id 200
set vlans pvlan400 pvlan isolation-vlan-id 500
set vlans pvlan100 interface xe-0/0/0.0 isolated
set vlans pvlan400 interface xe-0/0/0.0 isolated
set vlans comm600 interface xe-0/0/0.0
set vlans pvlan100 interface xe-0/0/2.0 isolated
set vlans pvlan400 interface xe-0/0/5.0 isolated
```

Step-by-Step Procedure

To configure the private VLANs and secondary VLAN trunk ports:

1. Configure the interfaces and port modes:

```
[edit interfaces]
user@switch# set xe-0/0/0 unit 0 family ethernet-switching port-mode trunk
user@switch# set xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
user@switch# set xe-0/0/2 unit 0 family ethernet-switching port-mode access
user@switch# set xe-0/0/3 unit 0 family ethernet-switching port-mode access
user@switch# set xe-0/0/5 unit 0 family ethernet-switching port-mode access
user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode access
```

2. Create the primary VLANs:

```
[edit vlans]
user@switch# set pvlan100 vlan-id 100
user@switch# set pvlan400 vlan-id 400
```

NOTE: Primary VLANs must always be tagged VLANs, even if they exist on only one device.

3. Configure the primary VLANs to be private:

```
[edit vlans]
user@switch# set pvlan100 pvlan
user@switch# set pvlan400 pvlan
```

4. Configure the PVLAN trunk port to carry the private VLAN traffic between the switches:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/1.0 pvlan-trunk
user@switch# set pvlan400 interface xe-0/0/1.0 pvlan-trunk
```

5. Create secondary VLAN comm300 with VLAN ID 300:

```
[edit vlans]
user@switch# set comm300 vlan-id 300
```

6. Configure the primary VLAN for comm300:

```
[edit vlans]
user@switch# set comm300 primary-vlan pvlan100
```

7. Configure the interface for comm300:

```
[edit vlans]
user@switch# set comm300 interface xe-0/0/3.0
```

8. Create secondary VLAN comm600 with VLAN ID 600:

```
[edit vlans]
user@switch# set comm600 vlan-id 600
```

9. Configure the primary VLAN for comm600:

```
[edit vlans]
user@switch# set comm600 primary-vlan pvlan400
```

10. Configure the interface for comm600:

```
[edit vlans]
user@switch# set comm600 interface xe-0/0/6.0
```

11. Configure the interswitch isolated VLANs:

```
[edit vlans]
user@switch# set pvlan100 pvlan isolation-vlan-id 200
user@switch# set pvlan400 pvlan isolation-vlan-id 500
```

NOTE: When you configure a secondary VLAN trunk port to carry an isolated VLAN, you must also configure an isolation-vlan-id. This is true even if the isolated VLAN exists only on one switch.

12. Enable trunk port xe-0/0/0 to carry secondary VLANs for the primary VLANs:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/0.0 isolated
user@switch# set pvlan400 interface xe-0/0/0.0 isolated
```

13. Configure trunk port xe-0/0/0 to carry comm600 (member of pvlan400):

```
[edit vlans]
user@switch# set comm600 interface xe-0/0/0.0
```

NOTE: You do not need to explicitly configure xe-0/0/0 to carry the isolated VLAN traffic (tags 200 and 500) because all the isolated ports in pvlan100 and pvlan400—including xe-0/0/0.0—are automatically included in the isolated VLANs created when you configured isolation-vlan-id 200 and isolation-vlan-id 500.

14. Configure xe-0/0/2 and xe-0/0/6 to be isolated:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/2.0 isolated
user@switch# set pvlan400 interface xe-0/0/5.0 isolated
```

Results

Check the results of the configuration on Switch 1:

```
[edit]
user@switch# show
interfaces {
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
```

```
        family ethernet-switching {
          port-mode trunk;
        }
      }
    }
  }
  vlans {
    comm300 {
      vlan-id 300;
        interface {
        xe-0/0/3.0;
      }
      primary-vlan pvlan100;
    }
    comm600 {
      vlan-id 600;
        interface {
        xe-0/0/6.0;
      }
      primary-vlan pvlan400;
    }
    pvlan100 {
      vlan-id 100;
      interface {
        xe-0/0/0.0;
        xe-0/0/2.0;
        xe-0/0/3.0;
        xe-0/0/1.0 {
          pvlan-trunk;
        }
      }
      no-local-switching;
      isolation-id 200;
    }
    pvlan400 {
      vlan-id 400;
      interface {
        xe-0/0/0.0;
        xe-0/0/5.0;
        xe-0/0/6.0;
        xe-0/0/1.0 {
          pvlan-trunk;
        }
      }
      no-local-switching;
      isolation-id 500;
    }
  }
```

Configuring the PVLANs on Switch 2

The configuration for Switch 2 is almost identical to the configuration for Switch 1. The most significant difference is that xe-0/0/0 on Switch 2 is configured as a promiscuous trunk port or a promiscuous access port, as Figure 12 on page 140 shows. In the following

configuration, xe-0/0/0 is configured as a promiscuous access port for primary VLAN pvlan100.

If traffic ingresses on VLAN-enabled port and egresses on a promiscuous access port, the VLAN tags are dropped on egress and the traffic is untagged at that point. For example, traffic for comm600 ingresses on the secondary VLAN trunk port configured on xe-0/0/0.0 on Switch 1 and carries tag 600 as it is forwarded through the secondary VLAN. When it egresses from xe-0/0/0.0 on Switch 2, it will be untagged if you configure xe-0/0/0.0 as a promiscuous access port as shown in this example. If you instead configure xe-0/0/0.0 as a promiscuous trunk port (port-mode trunk), the traffic for comm600 carries its primary VLAN tag (400) when it egresses.

CLI Quick Configuration

To quickly create and configure the PVLANs on Switch 2, copy the following commands and paste them into a switch terminal window:

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/3 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/5 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode access
set vlans pvlan100 vlan-id 100
set vlans pvlan400 vlan-id 400
set vlans pvlan100 pvlan
set vlans pvlan400 pvlan
set vlans pvlan100 interface xe-0/0/1.0 pvlan-trunk
set vlans pvlan400 interface xe-0/0/1.0 pvlan-trunk
set vlans comm300 vlan-id 300
set vlans comm300 primary-vlan pvlan100
set vlans comm300 interface xe-0/0/3.0
set vlans comm600 vlan-id 600
set vlans comm600 primary-vlan pvlan400
set vlans comm600 interface xe-0/0/6.0
set vlans pvlan100 pvlan isolation-vlan-id 200
set vlans pvlan400 pvlan isolation-vlan-id 500
set vlans pvlan100 interface xe-0/0/0.0 promiscuous
set vlans pvlan100 interface xe-0/0/2.0 isolated
set vlans pvlan400 interface xe-0/0/5.0 isolated
```

Step-by-Step Procedure

To configure the private VLANs and secondary VLAN trunk ports:

1. Configure the interfaces and port modes:

```
[edit interfaces]
user@switch# set xe-0/0/0 unit 0 family ethernet-switching port-mode access
user@switch# set xe-0/0/1 unit 0 family ethernet-switching port-mode trunk
user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan100
user@switch# set xe-0/0/1 unit 0 family ethernet-switching vlan members pvlan400
user@switch# set xe-0/0/2 unit 0 family ethernet-switching port-mode trunk
user@switch# set xe-0/0/3 unit 0 family ethernet-switching port-mode access
user@switch# set xe-0/0/5 unit 0 family ethernet-switching port-mode access
user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode access
```

2. Create the primary VLANs:

```
[edit vlans]
user@switch# set pvlan100 vlan-id 100
```

```
user@switch# set pvlan400 vlan-id 400
```

3. Configure the primary VLANs to be private:

```
[edit vlans]
user@switch# set pvlan100 pvlan
user@switch# set pvlan400 pvlan
```

4. Configure the PVLAN trunk port to carry the private VLAN traffic between the switches:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/1.0 pvlan-trunk
user@switch# set pvlan400 interface xe-0/0/1.0 pvlan-trunk
```

5. Create secondary VLAN comm300 with VLAN ID 300:

```
[edit vlans]
user@switch# set comm300 vlan-id 300
```

6. Configure the primary VLAN for comm300:

```
[edit vlans]
user@switch# set comm300 primary-vlan pvlan100
```

7. Configure the interface for comm300:

```
[edit vlans]
user@switch# set comm300 interface xe-0/0/3.0
```

8. Create secondary VLAN comm600 with VLAN ID 600:

```
[edit vlans]
user@switch# set comm600 vlan-id 600
```

9. Configure the primary VLAN for comm600:

```
[edit vlans]
user@switch# set comm600 primary-vlan pvlan400
```

10. Configure the interface for comm600:

```
[edit vlans]
user@switch# set comm600 interface xe-0/0/6.0
```

11. Configure the interswitch isolated VLANs:

```
[edit vlans]
user@switch# set pvlan100 pvlan isolation-vlan-id 200
user@switch# set pvlan400 pvlan isolation-vlan-id 500
```

12. Configure access port xe-0/0/0 to be promiscuous for pvlan100:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/0.0 promiscuous
```

NOTE: A promiscuous access port can be a member of only one primary VLAN.

13. Configure xe-0/0/2 and xe-0/0/6 to be isolated:

```
[edit vlans]
user@switch# set pvlan100 interface xe-0/0/2.0 isolated
user@switch# set pvlan400 interface xe-0/0/5.0 isolated
```

Results

Check the results of the configuration on Switch 2:

```
[edit]
user@switch# show
interfaces {
    xe-0/0/0 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
                vlan {
                    members pvlan100;
                }
            }
        }
    }
        xe-0/0/1 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members pvlan100;
                    members pvlan400;
                }
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
    xe-0/0/3 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/5 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
            family ethernet-switching {
                port-mode access;
            }
        }
    }
vlans {
    comm300 {
        vlan-id 300;
            interface {
```

```
        xe-0/0/3.0;
      }
      primary-vlan pvlan100;
    }
    comm600 {
      vlan-id 600;
        interface {
        xe-0/0/6.0;
      }
      primary-vlan pvlan400;
    }
    pvlan100 {
      vlan-id 100;
      interface {
        xe-0/0/0.0;
        xe-0/0/2.0;
        xe-0/0/3.0;
        xe-0/0/1.0 {
          pvlan-trunk;
        }
      }
      no-local-switching;
      isolation-id 200;
    }
    pvlan400 {
      vlan-id 400;
      interface {
        xe-0/0/5.0;
        xe-0/0/6.0;
        xe-0/0/1.0 {
          pvlan-trunk;
        }
      }
      no-local-switching;
      isolation-id 500;
    }
  }
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying That the Private VLAN and Secondary VLANs Were Created on page 149
- Verifying The Ethernet Switching Table Entries on page 150

Verifying That the Private VLAN and Secondary VLANs Were Created

Purpose Verify that the primary VLAN and secondary VLANs were properly created on Switch 1.

Action Use the **show vlans** command:

```
user@switch>  show vlans private-vlan

Name              Role        Tag      Interfaces
pvlan100          Primary     100      xe-0/0/0.0, xe-0/0/1.0, xe-0/0/2.0,
```

```
xe-0/0/3.0
__iso_pvlan100__    Isolated    200      xe-0/0/2.0
comm300             Community   300      xe-0/0/3.0

pvlan400            Primary     400      xe-0/0/0.0, xe-0/0/1.0, xe-0/0/5.0,
xe-0/0/6.0
__iso_pvlan400__    Isolated    500      xe-0/0/5.0
comm600             Community   600     xe-0/0/6.0
```

Meaning The output shows that the private VLANs were created and identifies the interfaces and secondary VLANs associated with them.

Verifying The Ethernet Switching Table Entries

Purpose Verify that the Ethernet switching table entries were created for primary VLAN pvlan100.

Action Show the Ethernet switching table entries for pvlan100.

```
user@switch> show ethernet-switching table vlan pvlan100 private-vlan
Ethernet-switching table: 0 unicast entries
  pvlan100              *                 Flood          - All-members
  pvlan100              00:10:94:00:00:02 Learn            xe-0/0/2.0
  __iso_pvlan100__      *                 Flood          - All-members
  __iso_pvlan100__      00:10:94:00:00:02 Replicated      - xe-0/0/2.0
```

Related Documentation

- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 37
- Understanding Private VLANs on page 28
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 33
- Creating a Private VLAN on a Single Switch on page 247
- Understanding Egress Firewall Filters with PVLANs on page 46
- *Troubleshooting Private VLANs*

CHAPTER 12

Q-in-Q Tunneling Configuration Example

Example: Setting Up Q-in-Q Tunneling

Service providers can use Q-in-Q tunneling to transparently pass Layer 2 VLAN traffic between customer sites without removing or changing the customer VLAN tags or class-of-service (CoS) settings. Data centers can use Q-in-Q tunneling to isolate customer traffic within a single site or when customer traffic flows between cloud data centers in different geographic locations.

This example describes how to set up Q-in-Q tunneling:

Requirements

This example requires one QFX Series device with Junos OS Release 12.1 or later.

Before you begin setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs on the neighboring switches. See "Configuring VLANs" on page 220.

Overview and Topology

In this service provider network, there are multiple customer VLANs mapped to one service VLAN.

Table 21 on page 151 lists the settings for the sample topology.

Table 21: Components of the Topology for Setting Up Q-in-Q Tunneling

Interface	Description
xe-0/0/11.0	Tagged S-VLAN trunk port
xe-0/0/12.0	Untagged customer-facing access port

Table 21: Components of the Topology for Setting Up Q-in-Q Tunneling *(continued)*

Interface	Description
xe-0/0/13.0	Untagged customer-facing access port
xe-0/0/14.0	Tagged S-VLAN trunk port

Configuration

CLI Quick Configuration

To quickly create and configure Q-in-Q tunneling, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans service-vlan vlan-id 1000
set vlans service-vlan dot1q-tunneling customer-vlans 1-100
set vlans service-vlan dot1q-tunneling customer-vlans 201-300
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members 1000
set interfaces xe-0/0/12 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members 1000
set interfaces xe-0/0/13 unit 0 family ethernet-switching port-mode access
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members 1000
set interfaces xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members 1000
set ethernet-switching-options dot1q-tunneling ether-type 0x9100
```

Step-by-Step Procedure

To configure Q-in-Q tunneling:

1. Set the VLAN ID for the S-VLAN:

   ```
   [edit vlans]
   user@switch# set service-vlan vlan-id 1000
   ```

2. Enable Q-in-Q tunneling and specify the customer VLAN ranges:

   ```
   [edit vlans]
   user@switch# set service-vlan dot1q-tunneling customer-vlans 1-100
   user@switch# set service-vlan dot1q-tunneling customer-vlans 201-300
   ```

3. Set the port mode and VLAN information for the interfaces:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/11 unit 0 family ethernet-switching vlan members 1000
   user@switch# set xe-0/0/12 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/12 unit 0 family ethernet-switching vlan members 1000
   user@switch# set xe-0/0/13 unit 0 family ethernet-switching port-mode access
   user@switch# set xe-0/0/13 unit 0 family ethernet-switching vlan members 1000
   user@switch# set xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
   user@switch# set xe-0/0/14 unit 0 family ethernet-switching vlan members 1000
   ```

4. Set the Q-in-Q Ethertype value (optional):

   ```
   [edit]
   user@switch# set ethernet-switching-options dot1q-tunneling ether-type 0x9100
   ```

Results

Check the results of the configuration:

```
user@switch> show configuration vlans service-vlan
vlan-id 1000 {
```

```
    dot1q-tunneling {
    customer-vlans [ 1-100 201-300 ];
}
user@switch> show configuration interfaces
xe-0/0/11 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 1000;
        }
    }
}
xe-0/0/12 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 1000;
        }
    }
}
xe-0/0/13 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 1000;
        }
    }
}
xe-0/0/14 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 1000;
        }
    }
}
user@switch> show ethernet-switching-options
dot1q-tunneling {
    ether-type 0x9100;
}
```

Verification

Confirm that the configuration is working properly.

Verifying That Q-in-Q Tunneling Was Enabled

Purpose Verify that Q-in-Q tunneling was properly enabled.

Action Use the **show vlans** command:

```
user@switch>  show vlans service-vlan extensive
VLAN: service-vlan, Created at: Wed Mar 14 07:17:53 2012
802.1Q Tag: 1000, Internal index: 18, Admin State: Enabled, Origin: Static
Dot1q Tunneling Status: Enabled
Customer VLAN ranges:
```

```
                    1-100
                    201-300
Protocol: Port Mode
Number of interfaces: Tagged 2 (Active = 0), Untagged  2 (Active = 0)
      xe-0/0/11.0, tagged, trunk
      xe-0/0/14.0, tagged, trunk
      xe-0/0/12.0, untagged, access
      xe-0/0/13.0, untagged, access
```

Meaning The output indicates that Q-in-Q tunneling is enabled and that the VLAN is tagged and shows the associated customer VLANs.

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 61
- Configuring Q-in-Q Tunneling on page 253
- *Troubleshooting Q-in-Q and VLAN Translation Configuration*

CHAPTER 13

Reflective Relay Configuration Example

Example: Configuring Reflective Relay for Use with VEPA Technology

Reflective relay must be configured on a switch that receives virtual machine aggregated packets, such as Virtual Ethernet Port Aggregator (VEPA) packets, because some of these packets might be sent back to the server destined for another virtual machine on the same server. Reflective relay returns those packets to the original device using the same downstream port that delivered the packets to the switch.

NOTE: This example uses Junos OS for QFX3500 and QFX3600 switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see *Example: Configuring Reflective Relay for Use with VEPA Technology*.

This example shows how to configure a switch port interface to return packets sent by VEPA on the downstream interface back to the server using the same downstream interface:

Requirements

This example uses the following hardware and software components:

- One QFX3500 switch
- One server
- Junos OS Release 12.1 or later for the QFX Series

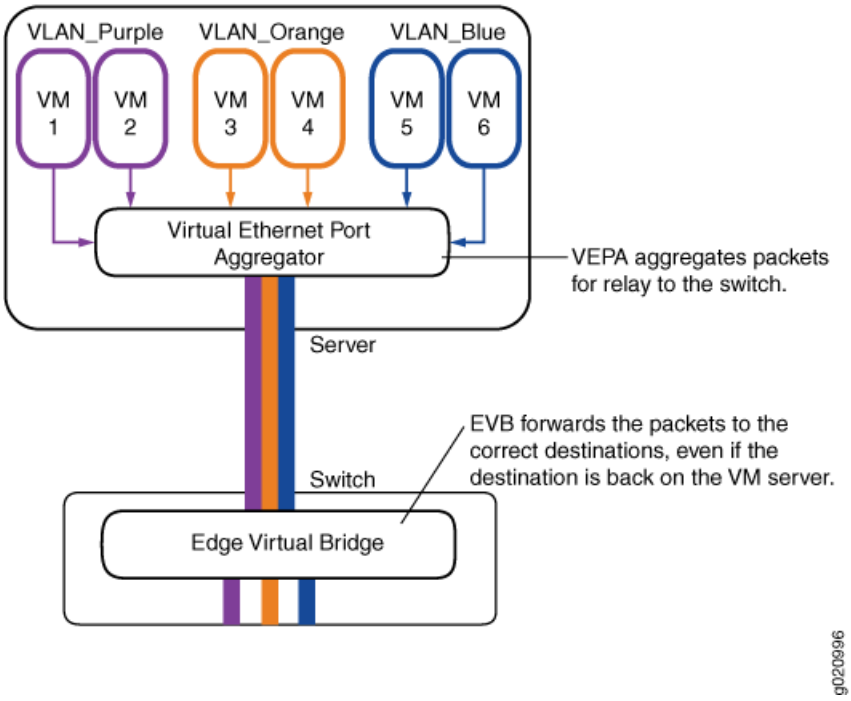
Before you configure reflective relay on a switch port, be sure you have:

- Configured a server with six virtual machines, VM 1 through VM 6.
- Configured the server with three VLANS named VLAN_Purple, VLAN_Orange, and VLAN_Blue and added two virtual machines to each VLAN.
- Configured the same three VLANs named VLAN_Purple, VLAN_Orange, and VLAN_Blue on one interface.
- Installed and configured VEPA to aggregate the virtual machine packets.

Overview and Topology

In this example, illustrated in Figure 13 on page 157, a switch is connected to one server that is hosting six virtual machines and is configured with a VEPA for aggregating packets. The server's six virtual machines are VM 1 through VM 6, and each virtual machine belongs to one of the three server VLANs, VLAN_Purple, VLAN_Orange, or VLAN_Blue. Instead of the server directly passing packets between virtual machines, packets from any of the three VLANS that are destined for another one of the three VLANs are aggregated using VEPA technology and passed to the switch for processing. You must configure the switch port to accept these aggregated packets on the downstream interface and to return appropriate packets to the server on the same downstream interface after they are processed. Figure 13 on page 157 shows the topology for this example.

Figure 13: Reflective Relay Topology



In this example, you configure the physical Ethernet switch port interface for tagged-access port mode and reflective relay. Configuring tagged-access port mode allows the interface to accept VLAN tagged packets. Configuring reflective relay allows the downstream port to return those packets on the same interface. Table 22 on page 157 shows the components used in this example.

Table 22: Components of the Topology for Configuring Reflective Relay

Component	Description
QFX3500 switch	Switch that supports reflective relay. For a list of switches that support this feature, see *QFX Series Software Features*.
xe-0/0/2	Switch interface to the server.
Server	Server with virtual machines and VEPA technology.
Virtual machines	Six virtual machines located on the server: V1, V2, V3, V4, V5, and V6.
VLANs	Three VLANs: VLAN_Purple, VLAN_Orange, and VLAN_Blue. Each VLAN has two virtual machine members.
VEPA	Virtual Ethernet port aggregator that aggregates virtual machine packets on the server before the resulting single stream is transmitted to the switch.

Configuration

To configure reflective relay, perform these tasks:

- Configuring Reflective Relay on the Port on page 158

Configuring Reflective Relay on the Port

CLI Quick Configuration

To quickly configure reflective relay, copy the following commands and paste them into the switch window:

```
[edit]
set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Blue VLAN_Orange VLAN_Purple]
```

Step-by-Step Procedure

To configure reflective relay:

1. Configure the tagged-access port mode on the interface:

   NOTE: Configure the port mode as tagged-access otherwise you will receive an error when you commit the configuration.

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Configure reflective relay on the interface to allow it to both accept and send packets:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
   ```

3. Configure the interface for the three VLANs on the server:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members [VLAN_Purple VLAN_Orange VLAN_Blue]
   ```

Results

Check the results of the configuration:

```
[edit interfaces xe-0/0/2]
user@switch# show
unit 0 {
    family ethernet-switching {
    port-mode tagged-access;
    reflective-relay;
    vlan {
     members [ VLAN_Purple VLAN_Orange VLAN_Blue ];
    }
  }
}
```

Verification

To confirm that reflective relay is enabled and working correctly, perform these tasks:

- Verifying That Reflective Relay Is Enabled and Working Correctly on page 159

Verifying That Reflective Relay Is Enabled and Working Correctly

Purpose Verify that reflective relay is enabled and working correctly.

Action Use the **show ethernet-switching interfaces detail** command to display the reflective relay status:

```
user@switch> show ethernet-switching interfaces xe-0/0/2 detail
Interface: xe-0/0/2, Index: 66, State: down, Port mode: Tagged-access
Reflective Relay Status: Enabled
Ether type for the interface: 0x8100
VLAN membership:
    VLAN_Purple, 802.1Q Tag: 450, tagged, unblocked
    VLAN_Orange, 802.1Q Tag: 460, tagged, unblocked
    VLAN_Blue, 802.1Q Tag: 470, tagged, unblocked
Number of MACs learned on IFL: 0
```

Confirm that reflective relay is working by sending a Layer 2 broadcast message from one virtual machine to another virtual machine located on the same VLAN. Check the switch to verify that the switch sends the packets back on the same interface on which they were received. One way to check this is to set up port mirroring on the switch interface, connect a traffic generator to the mirrored interface, and use the traffic generator to examine packets.

Alternatively, if you do not have a traffic generator available, you can send traffic between two virtual machines with FTP, Telnet, or SSH, while running the **tcpdump** utility on the receiver virtual machine port to capture reflected packets.

Meaning The reflective relay status is **Enabled**, meaning that interface **xe-0/0/2** is configured for the tagged-access port mode, which accepts VLAN-tagged packets, and for reflective relay, which accepts and returns packets on the same interface.

When the traffic generator shows packets arriving at the switch and returning to the server on the same interface, reflective relay is working.

Related Documentation

- Understanding Reflective Relay for Use with VEPA Technology on page 27
- Configuring Reflective Relay on page 255
- *Configuring Port Mirroring*
- Configuring VLANs on page 220
- **port-mode on page 359**
- **reflective-relay on page 313**

CHAPTER 14

VLAN Configuration Examples

Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations

The QFX Series products provide Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP). Configure BPDU protection on interfaces to prevent them from receiving BPDUs that could result in STP misconfigurations, which could lead to network outages.

This example describes how to configure BPDU protection on access interfaces in QFX Series products in an RSTP topology:

Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for the QFX Series
- Two edged-linked switches in an RSTP topology

NOTE: By default, RSTP is enabled on the QFX Series.

Overview and Topology

A loop-free network is supported through the exchange of a special type of frame called a bridge protocol data unit (BPDU). However, receipt of BPDUs on certain interfaces in an STP, RSTP, or MSTP topology. It can lead to network outages by triggering an STP misconfiguration. To prevent such outages, enable BPDU protection on those interfaces that should not receive BPDUs.

Enable BPDU protection on switch interfaces connected to user devices or on interfaces on which no BPDUs are expected, such as edge ports. If a BPDU is received on a BPDU-protected interface, the interface is disabled and stops forwarding frames.

Two switches are displayed in Figure 14 on page 162. In this example, Switch 1 and Switch 2 are configured for RSTP and create a loop-free topology. The interfaces on Switch 2 are access ports.

This example shows you how to configure interface **xe-0/0/5** and interface **xe-0/0/6** as edge ports and how to configure BPDU protection. When BPDU protection is enabled, the interfaces transition to a blocking state when they receive BPDUs.

Figure 14: BPDU Protection Topology

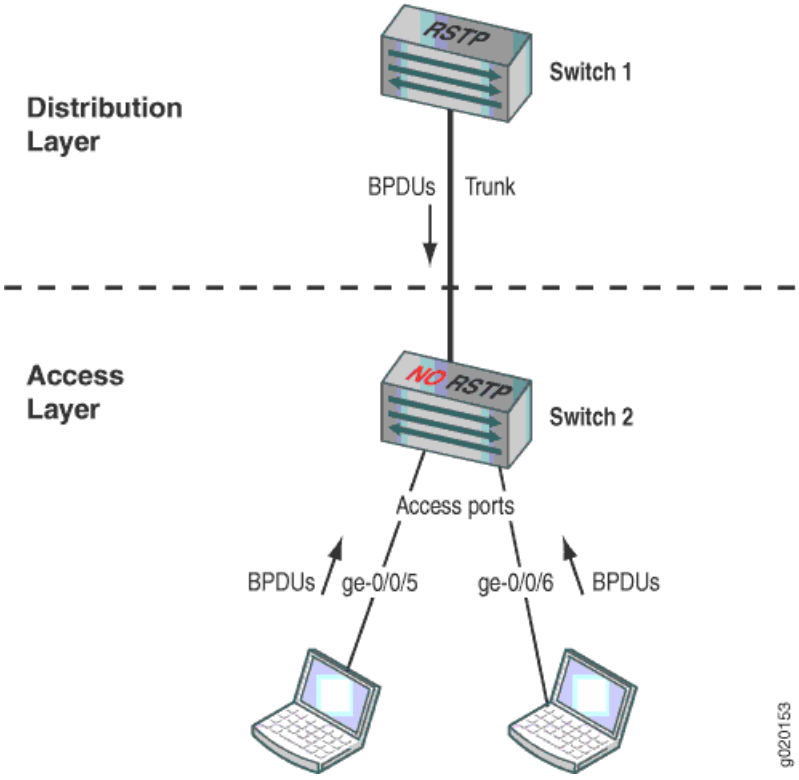


Table 23 on page 163 shows the components that will be configured for BPDU protection.

Table 23: Components of the Topology for Configuring BPDU Protection on the QFX Series

Component	Settings
Switch 1 (Distribution Layer)	Switch 1 is connected to Switch 2 on a trunk interface.
Switch 2 (Access Layer)	Switch 2 has these access ports that require BPDU protection: • xe-0/0/5 • xe-0/0/6

This configuration example uses an RSTP topology. You also can configure BPDU protection for STP or MSTP topologies at the [**edit protocols** (**mstp** | **stp**)] hierarchy level.

Configuration

CLI Quick Configuration

To quickly configure BPDU protection on Switch 2, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols rstp interface xe-0/0/5 edge
set protocols rstp interface xe-0/0/6 edge
set protocols rstp bpdu-block-on-edge
```

Step-by-Step Procedure

To configure BPDU protection:

1. Configure interface **xe-0/0/5** and interface **xe-0/0/6** on Switch 2 as edge ports:

   ```
   [edit protocols rstp]
   user@switch# set interface xe-0/0/5 edge
   user@switch#set interface xe-0/0/6 edge
   ```

2. Configure BPDU protection on all edge ports:

   ```
   [edit protocols rstp]
   user@switch# setbpdu-block-on-edge
   ```

Results

Check the results of the configuration:

```
user@switch> show configuration protocols rstp
interface xe-0/0/5.0 {
   edge;
}
interface xe-0/0/6.0 {
   edge;
}
bpdu-block-on-edge;
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying the Interface State Before BPDU Protection Is Triggered on page 164
- Verifying That BPDU Protection Is Working Correctly on page 164

Displaying the Interface State Before BPDU Protection Is Triggered

Purpose Before BPDUs are being received from the devices connected to interface **xe-0/0/5** and interface **xe-0/0/6**, confirm the interface state.

Action You can verify the interface state using the **show spanning-tree interface** command:

```
user@switch> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/0.0     128:513      128:513  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/1.0     128:514      128:514  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/2.0     128:515      128:515  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/3.0     128:516      128:516  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/4.0     128:517      128:517  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/5.0     128:518      128:518  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/6.0     128:519      128:519  32768.0019e2503f00     20000  FWD    DESG
[output truncated]
```

Meaning The output shows that interface **xe-0/0/5.0** and interface **xe-0/0/6.0** are designated ports in a forwarding state.

Verifying That BPDU Protection Is Working Correctly

Purpose In this example, the devices connected to Switch 2 start sending BPDUs to interface **xe-0/0/5.0** and interface **xe-0/0/6.0** . Verify that BPDU protection is configured on the interfaces.

Action You can verify that BPDU protection is configured on the interfaces by using the **show spanning-tree interface** command:

```
user@switch>  show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/0.0     128:513      128:513  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/1.0     128:514      128:514  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/2.0     128:515      128:515  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/3.0     128:516      128:516  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/4.0     128:517      128:517  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/5.0     128:518      128:518  32768.0019e2503f00     20000  BLK    DIS
(Bpdu-Incon)
xe-0/0/6.0     128:519      128:519  32768.0019e2503f00     20000  BLK    DIS
(Bpdu-Incon)
xe-0/0/7.0     128:520        128:1  16384.00aabbcc0348     20000  FWD    ROOT
xe-0/0/8.0     128:521      128:521  32768.0019e2503f00     20000  FWD    DESG
[output truncated]
```

Meaning When BPDUs are sent from the devices to interface **xe-0/0/5.0** and interface **xe-0/0/6.0** on Switch 2, the output from the operational mode command **show spanning-tree interface** shows that the interfaces have transitioned to a BPDU inconsistent state. The BPDU inconsistent state blocks the interfaces and prevents them from forwarding traffic.

Disabling the BPDU protection configuration on an interface does not unblock the interface. If the **disable-timeout** statement has been included in the BPDU configuration, the interface automatically returns to service after the timer expires. Otherwise, use the operational mode command **clear bpdu-error** to unblock the interface.

If the devices connected to Switch 2 send BPDUs to the interfaces again, BPDU protection is triggered once more and the interfaces transition back to the BPDU inconsistent state. In such cases, you need to find and repair the misconfiguration on the devices that is triggering the sending of BPDUs to Switch 2.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Example: Configuring Root Protection to Enforce Root Bridge Placement in Spanning Trees on page 207
- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55

Example: Configuring Faster Convergence and Improving Network Stability with RSTP

The QFX Series products use Rapid Spanning Tree Protocol (RSTP) to provide a loop-free topology. RSTP identifies certain links as point to point. When a point-to-point link fails, the alternate link can transition to the forwarding state. RSTP provides quicker

reconvergence time than original STP because it uses protocol handshake messages rather than fixed timeouts. Eliminating the need to wait for timers to expire makes RSTP more efficient than STP.

This example describes how to configure RSTP on four QFX3500 switches:

Requirements

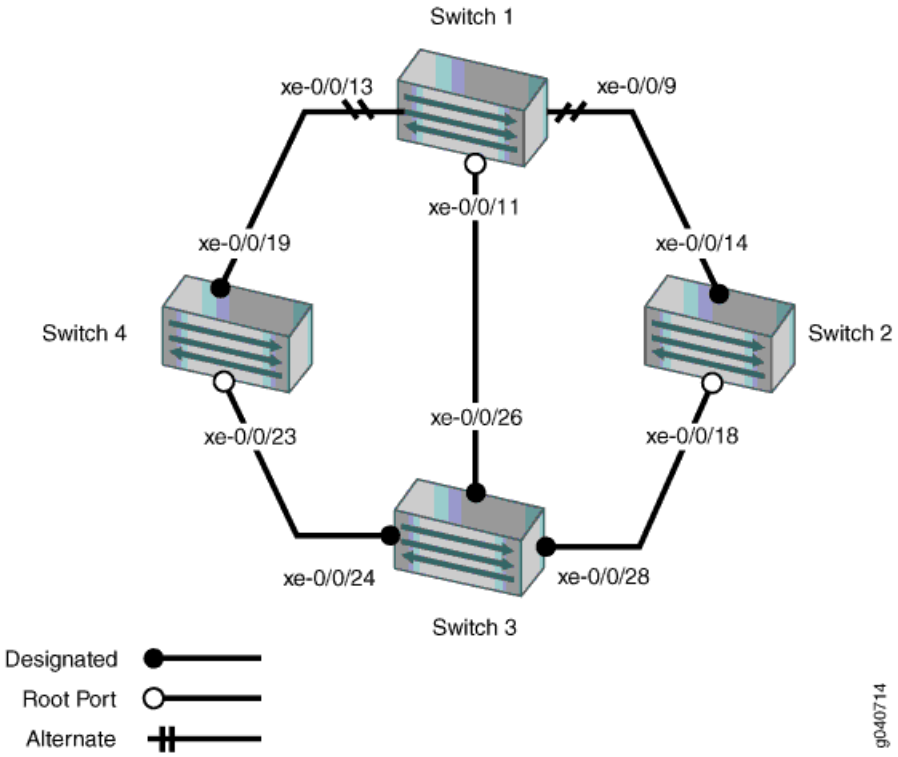
This example uses the following hardware and software components:

- Junos OS Release 11.1 for the QFX3500 switches
- Four QFX3500 switches

Overview and Topology

In this example, QFX3500 switches are connected in the topology displayed in Figure 15 on page 167 to create a loop-free topology.

Figure 15: Network Topology for RSTP



The interfaces shown in Table 24 on page 167 will be configured for RSTP.

NOTE: You can configure RSTP on logical or physical interfaces. This example shows RSTP configured on logical interfaces.

Table 24: Topology for Configuring RSTP on the QFX Series

Components	Settings
Switch 1	The following ports on Switch 1 are connected in this way: • **xe-0/0/9** is connected to Switch 2 • **xe-0/0/13** is connected to Switch 4 • **xe-0/0/11** is connected to Switch 3
Switch 2	The following ports on Switch 2 are connected in this way: • **xe-0/0/14** is connected to Switch 1 • **xe-0/0/18** is connected to Switch 3
Switch 3	The following ports on Switch 3 are connected in this way: • **xe-0/0/26** is connected to Switch 1 • **xe-0/0/28** is connected to Switch 2 • **xe-0/0/24** is connected to Switch 4

Table 24: Topology for Configuring RSTP on the QFX Series *(continued)*

Components	Settings
Switch 4	The following ports on Switch 4 are connected in this way: • **xe-0/0/19** is connected to Switch 1 • **xe-0/0/23** is connected to Switch 3
VLAN names and tag IDs	**sales-vlan**, tag **10** **engineering-vlan**, tag **20** **publications-vlan**, tag **30** **support-vlan**, tag **40**

This configuration example creates a loop-free topology between four switches using RSTP.

An RSTP topology contains ports that have specific roles:

- The *root port* is responsible for forwarding data to the root bridge.
- The *alternate port* is a standby port for the root port. When a root port goes down, the alternate port becomes the active root port.
- The *designated port* forwards data to the downstream network segment or device.
- The *backup port* is a backup port for the designated port. When a designated port goes down, the backup port becomes the active designated port and starts forwarding data.

Configuring RSTP on Switch 1

CLI Quick Configuration

To quickly configure interfaces and RSTP on Switch 1, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the interface-mode statement instead of the port-mode statement. The port-mode statement has been replaced with the interface-mode statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/9 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/13 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/9 unit 0 family ethernet-switching port-mode trunk
```

```
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
set protocols rstp bridge-priority 16k
set protocols rstp interface xe-0/0/13.0 cost 1000
set protocols rstp interface xe-0/0/13.0 mode point-to-point
set protocols rstp interface xe-0/0/9.0 cost 1000
set protocols rstp interface xe-0/0/9.0 mode point-to-point
set protocols rstp interface xe-0/0/11.0 cost 1000
set protocols rstp interface xe-0/0/11.0 mode point-to-point
```

Step-by-Step Procedure

To configure interfaces and RSTP on Switch 1:

1. Configure the VLANs **sales-vlan**, **engineering-vlan** and **publications-vlan**, and **support-vlan**:

```
[edit vlans]
user@switch1# set sales-vlan description “Sales VLAN”
user@switch1# set sales-vlan vlan-id 10
user@switch1# set engineering-vlan description “Engineering VLAN”
user@switch1# set engineering-vlan vlan-id 20
user@switch1# set publications-vlan description “Publications VLAN”
user@switch1# set publications-vlan vlan-id 30
```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch1# set xe-0/0/13 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch1# set xe-0/0/9 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch1# set xe-0/0/11 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit interfaces]
user@switch1# set xe-0/0/13 unit 0 family ethernet-switching port-mode trunk
user@switch1# set xe-0/0/9 unit 0 family ethernet-switching port-mode trunk
user@switch1# set xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
```

4. Configure RSTP on the switch:

```
[edit protocols]
user@switch1# rstp bridge-priority 16k
user@switch1# rstp interface xe-0/0/13.0 cost 1000
user@switch1# rstp interface xe-0/0/13.0 mode point-to-point
user@switch1# rstp interface xe-0/0/9.0 cost 1000
user@switch1# rstp interface xe-0/0/9.0 mode point-to-point
user@switch1# rstp interface xe-0/0/11.0 cost 1000
user@switch1# rstp interface xe-0/0/11.0 mode point-to-point
```

Results

Check the results of the configuration:

```
user@switch1> show configuration
```

```
interfaces {
    xe-0/0/13 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
    xe-0/0/9 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
    xe-0/0/11 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
}
protocols {
    rstp {
        bridge-priority 16k;
            interface xe-0/0/13.0 {
                cost 1000;
                mode point-to-point;
            }
            interface xe-0/0/9.0 {
                cost 1000;
                mode point-to-point;
            }
            interface xe-0/0/11.0 {
                cost 1000;
                mode point-to-point;
            }
        }
    }
}
vlans {
    sales-vlan {
        vlan-id 10;
    }
    engineering-vlan {
        vlan-id 20;
```

```
    }
    publications-vlan {
      vlan-id 30;
    }
    support-vlan {
      vlan-id 40;
    }
  }
```

Configuring RSTP on Switch 2

CLI Quick Configuration

To quickly configure interfaces and RSTP on Switch 2, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/18 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/18 unit 0 family ethernet-switching port-mode trunk
set protocols rstp bridge-priority 32k
set protocols rstp interface xe-0/0/14.0 cost 1000
set protocols rstp interface xe-0/0/14.0 mode point-to-point
set protocols rstp interface xe-0/0/18.0 cost 1000
set protocols rstp interface xe-0/0/18.0 mode point-to-point
```

Step-by-Step Procedure

To configure interfaces and RSTP on Switch 2:

1. Configure the VLANs **sales-vlan**, **engineering-vlan** and **publications-vlan**, and **support-vlan**:

   ```
   [edit vlans]
   user@switch2# set sales-vlan description “Sales VLAN”
   user@switch2# set sales-vlan vlan-id 10
   user@switch2# set engineering-vlan description “Engineering VLAN”
   user@switch2# set engineering-vlan vlan-id 20
   user@switch2# set publications-vlan description “Publications VLAN”
   user@switch2# set publications-vlan vlan-id 30
   user@switch2# set support-vlan vlan-description “Support VLAN”
   user@switch2# set publications-vlan vlan-id 40
   ```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch2# set xe-0/0/14 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch2# set xe-0/0/18 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the interface-mode statement instead of the port-mode statement. The port-mode statement has been replaced with the interface-mode statement.

```
[edit interfaces]
user@switch2# set xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
user@switch2# set xe-0/0/18 unit 0 family ethernet-switching port-mode trunk
```

4. Configure RSTP on the switch:

```
[edit protocols]
user@switch2# rstp bridge-priority 32k
user@switch2# rstp interface xe-0/0/14.0 cost 1000
user@switch2# rstp interface xe-0/0/14.0 mode point-to-point
user@switch2# rstp interface xe-0/0/18.0 cost 1000
user@switch2# rstp interface xe-0/0/18.0 mode point-to-point
```

Results Check the results of the configuration:

```
user@switch2> show configuration
interfaces {
    xe-0/0/14 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
    xe-0/0/18 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
}
protocols {
    rstp {
        bridge-priority 32k;
            interface xe-0/0/14.0 {
                cost 1000;
```

```
            mode point-to-point;
          }
          interface xe-0/0/18.0 {
            cost 1000;
            mode point-to-point;
          }
        }
      }
    }
    vlans {
      sales-vlan {
        vlan-id 10;
      }
      engineering-vlan {
        vlan-id 20;
      }
      publications-vlan {
        vlan-id 30;
      }
      support-vlan {
        vlan-id 40;
      }
    }
```

Configuring RSTP on Switch 3

CLI Quick Configuration

To quickly configure interfaces and RSTP on Switch 3, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/28 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/26 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/28 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/24 unit 0 family ethernet-switching port-mode trunk
set protocols rstp bridge-priority 8k
set protocols rstp interface xe-0/0/26.0 cost 1000
set protocols rstp interface xe-0/0/26.0 mode point-to-point
set protocols rstp interface xe-0/0/28.0 cost 1000
```

```
set protocols rstp interface xe-0/0/28.0 mode point-to-point
set protocols rstp interface xe-0/0/24.0 cost 1000
set protocols rstp interface xe-0/0/24.0 mode point-to-point
```

Step-by-Step Procedure

To configure interfaces and RSTP on Switch 3:

1. Configure the VLANs **sales-vlan**, **engineering-vlan**, **publications-vlan**, and **support-vlan**:

```
[edit vlans]
user@switch3# set sales-vlan description “Sales VLAN”
user@switch3# set sales-vlan vlan-id 10
user@switch3# set engineering-vlan description “Engineering VLAN”
user@switch3# set engineering-vlan vlan-id 20
user@switch3# set publications-vlan description “Publications VLAN”
user@switch3# set publications-vlan vlan-id 30
user@switch3# set support-vlan description “Support VLAN”
user@switch3# set publications-vlan vlan-id 40
```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch3# set xe-0/0/26 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch3# set xe-0/0/28 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch3# set xe-0/0/24 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit interfaces]
user@switch3# set xe-0/0/26 unit 0 family ethernet-switching port-mode trunk
user@switch3# set xe-0/0/28 unit 0 family ethernet-switching port-mode trunk
user@switch3# set xe-0/0/24 unit 0 family ethernet-switching port-mode trunk
```

4. Configure RSTP on the switch:

```
[edit protocols]
user@switch3# rstp bridge-priority 8k
user@switch3# rstp interface xe-0/0/26.0 cost 1000
user@switch3# rstp interface xe-0/0/26.0 mode point-to-point
user@switch3# rstp interface xe-0/0/28.0 cost 1000
user@switch3# rstp interface xe-0/0/28.0 mode point-to-point
user@switch3# rstp interface xe-0/0/24.0 cost 1000
user@switch3# rstp interface xe-0/0/24.0 mode point-to-point
```

Results

Check the results of the configuration:

```
user@switch3> show configuration
  interfaces {
    xe-0/0/26 {
      unit 0 {
        family ethernet-switching {
```

```
            port-mode trunk;
            vlan {
              members [10 20 30 40];
            }
          }
        }
      }
      xe-0/0/28 {
        unit 0 {
          family ethernet-switching {
            port-mode trunk;
            vlan {
              members [10 20 30 40];
            }
          }
        }
      }
      xe-0/0/24 {
        unit 0 {
          family ethernet-switching {
            port-mode trunk;
            vlan {
              members [10 20 30 40];
            }
          }
        }
      }
    }
  }
  protocols {
    rstp {
      bridge-priority 8k;
        interface xe-0/0/26.0 {
          cost 1000;
          mode point-to-point;
        }
        interface xe-0/0/28.0 {
          cost 1000;
          mode point-to-point;
        }
        interface xe-0/0/24.0 {
          cost 1000;
          mode point-to-point;
        }
      }
      bridge-priority 8k;
        }
      }
    }
  }
  vlans {
    sales-vlan {
      vlan-id 10;
    }
    engineering-vlan {
      vlan-id 20;
```

```
    }
    publications-vlan {
      vlan-id 30;
    }
    support-vlan {
      vlan-id 40;
    }
  }
```

Configuring RSTP on Switch 4

CLI Quick Configuration

To quickly configure interfaces and RSTP on Switch 4, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan–id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/23 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/19 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/23 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/19 unit 0 family ethernet-switching port-mode trunk
set protocols rstp bridge-priority 16k
set protocols rstp interface xe-0/0/23.0 cost 1000
set protocols rstp interface xe-0/0/23.0 mode point-to-point
set protocols rstp interface xe-0/0/19.0 cost 1000
set protocols rstp interface xe-0/0/19.0 mode point-to-point
```

Step-by-Step Procedure

To configure interfaces and RSTP on Switch 4:

1. Configure the VLANs **sales-vlan**, **engineering-vlan**, **publications-vlan**, and **support-vlan**:

   ```
   [edit vlans]
   user@switch4# set sales-vlan description “Sales VLAN”
   user@switch4# set sales-vlan vlan-id 10
   user@switch4# set engineering-vlan description “Engineering VLAN”
   user@switch4# set engineering-vlan vlan-id 20
   user@switch4# set publications-vlan description “Publications VLAN”
   user@switch4# set publications-vlan vlan-id 30
   user@switch4# set support-vlan description “Support VLAN”
   user@switch4# set publications-vlan vlan-id 40
   ```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

   ```
   [edit interfaces]
   ```

```
user@switch4# set xe-0/0/23 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch4# set xe-0/0/19 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring RSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the interface-mode statement instead of the port-mode statement. The port-mode statement has been replaced with the interface-mode statement.

```
[edit interfaces]
user@switch4# set xe-0/0/23 unit 0 family ethernet-switching port-mode trunk
user@switch4# set xe-0/0/19 unit 0 family ethernet-switching port-mode trunk
```

4. Configure RSTP on the switch:

```
[edit protocols]
user@switch4# rstp bridge-priority 16k
user@switch4# rstp interface all cost 1000
user@switch4# rstp interface xe-0/0/23.0 cost 1000
user@switch4# rstp interface xe-0/0/23.0 mode point-to-point
user@switch4# rstp interface xe-0/0/19.0 cost 1000
user@switch4# rstp interface xe-0/0/19.0 mode point-to-point
```

Results Check the results of the configuration:

```
user@switch4> show configuration
interfaces {
    xe-0/0/23 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
    xe-0/0/19 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members [10 20 30 40];
                }
            }
        }
    }
}
protocols {
    rstp {
        bridge-priority 16k;
            interface xe-0/0/23.0 {
                cost 1000;
```

```
            mode point-to-point;
          }
          interface xe-0/0/19.0 {
            cost 1000;
            mode point-to-point;
          }
        }
      }
    }
    vlans {
      sales-vlan {
        vlan-id 10;
      }
      engineering-vlan {
        vlan-id 20;
      }
      publications-vlan {
        vlan-id 30;
      }
      support-vlan {
        vlan-id 40;
      }
    }
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying RSTP Configuration on Switch 1 on page 178
- Verifying RSTP Configuration on Switch 2 on page 179
- Verifying RSTP Configuration on Switch 3 on page 179
- Verifying RSTP Configuration on Switch 4 on page 179

Verifying RSTP Configuration on Switch 1

Purpose Verify that the RSTP configuration on Switch 1 is correct.

Action In operational mode, issue the **show spanning-tree interface** command:

```
user@switch1> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID         bridge ID          Cost
xe-0/0/13.0    128:527      128:525  16384.0019e25040e0      1000  BLK    ALT
xe-0/0/9.0     128:529      128:513  32768.0019e2503d20      1000  BLK    ALT
xe-0/0/11.0    128:531      128:513   8192.0019e25051e0      1000  FWD    ROOT
```

Meaning See the topology in Figure 15 on page 167. The operational mode command **show spanning-tree interface** shows that **xe-0/0/13.0** is in a forwarding state. The other interfaces on Switch 1 are blocked.

Verifying RSTP Configuration on Switch 2

Purpose Verify that the RSTP configuration on Switch 2 is correct.

Action In operational mode issue the **show spanning-tree interface** command:

```
user@switch2> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/14.0     128:513      128:513  32768.0019e2503d20      1000  BLK    DESG
xe-0/0/18.0     128:519      128:515   8192.0019e25051e0      1000  FWD    ROOT
```

Meaning See the topology in Figure 15 on page 167. The operational mode command **show spanning-tree interface** shows that interface **xe-0/0/18.0** is in a forwarding state and the root port. The other interface on Switch 2 is blocked.

Verifying RSTP Configuration on Switch 3

Purpose Verify that the RSTP configuration on Switch 3 is correct.

Action In operational mode, issue the **show spanning-tree interface** command:

```
user@switch3> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/26.0     128:513      128:513   8192.0019e25051e0      1000  FWD    DESG
xe-0/0/28.0     128:515      128:515   8192.0019e25051e0      1000  FWD    DESG
xe-0/0/24.0     128:517      128:517   8192.0019e25051e0      1000  FWD    DESG
```

Meaning See the topology in Figure 15 on page 167. The operational mode command **show spanning-tree interface** shows that no interface is the root interface.

Verifying RSTP Configuration on Switch 4

Purpose Verify the RSTP configuration on Switch 4.

Action In operational mode, issue the **show spanning-tree interface** command:

```
user@switch4> show spanning-tree interface
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost

xe-0/0/23.0    128:523      128:517   8192.0019e25051e0      1000  FWD    ROOT
xe-0/0/19.0    128:525      128:525  16384.0019e25040e0      1000  FWD    DESG
```

Meaning See the topology in Figure 15 on page 167. The operational mode command **show spanning-tree interface** shows that interface **xe-0/0/23.0** is the root interface and is in the forwarding state.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding RSTP on page 53

Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree

The QFX Series products provide Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP). Loop protection increases the efficiency of STP, RSTP, and MSTP by preventing interfaces from moving into a forwarding state that would create a loop in the network.

This example describes how to configure loop protection for an interface for the QFX Series in an RSTP topology:

- Requirements on page 180
- Overview and Topology on page 181
- Configuration on page 182
- Verification on page 182

Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for the QFX Series
- Three switches in an RSTP topology

NOTE: **By default, RSTP is enabled for the QFX Series.**

Overview and Topology

A loop-free network in spanning-tree topologies is supported through the exchange of a special type of frame called a bridge protocol data unit (BPDU). Peer STP applications running on the switch interfaces use BPDUs to communicate. Ultimately, the exchange of BPDUs determines which interfaces block traffic (preventing loops) and which interfaces become root ports and forward traffic.

A blocking interface can transition to the forwarding state in error if the interface stops receiving BPDUs from its designated port on the segment. Such a transition error can occur when there is a hardware error on the switch or software configuration error between the switch and its neighbor. When this happens, a loop appears in the spanning tree. Loops in a Layer 2 topology cause broadcast, unicast, and multicast frames to continuously circle the looped network. As a switch processes a flood of frames in a looped network, its resources become depleted, and the ultimate result is a network outage.

NOTE: **An interface can be configured for either loop protection or root protection, but not for both.**

Three switches are displayed in Figure 16 on page 181. In this example, they are configured for RSTP and create a loop-free topology. Interface **xe-0/0/6** is blocking traffic between Switch 3 and Switch 1; thus, traffic is forwarded through interface **xe-0/0/7** on Switch 2. BPDUs are being sent from the root bridge on Switch 1 to both of these interfaces.

This example shows how to configure loop protection on interface **xe-0/0/6** to prevent it from transitioning from a blocking state to a forwarding state and creating a loop in the spanning-tree topology.

Figure 16: Network Topology for Loop Protection

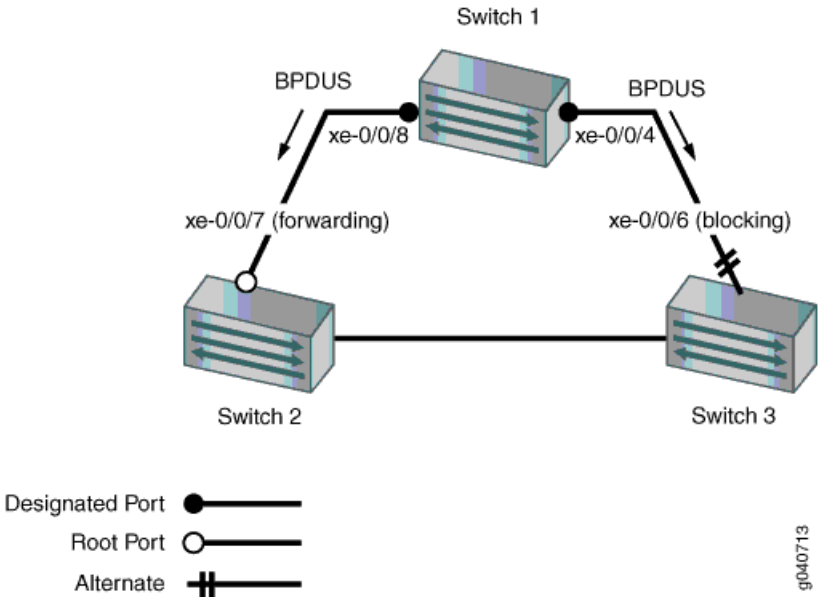


Table 25 on page 182 shows the components that will be configured for loop protection.

Table 25: Topology for Configuring Loop Protection on the QFX Series

Components	Settings
Switch 1	Switch 1 is the root bridge.
Switch 2	Switch 2 has the root port **xe-0/0/7**.
Switch 3	Switch 3 is connected to Switch 1 through interface **xe-0/0/6**.

A spanning-tree topology contains ports that have specific roles:

- The *root port* is responsible for forwarding data to the root bridge.
- The *alternate port* is a standby port for the root port. When a root port goes down, the alternate port becomes the active root port.
- The *designated port* forwards data to the downstream network segment or device.

This configuration example uses an RSTP topology. However, you can also configure loop protection for STP or MSTP topologies at the [**edit protocols** (**mstp** | **stp**)] hierarchy level.

Configuration

CLI Quick Configuration

To quickly configure loop protection on interface **xe-0/0/6**:

```
[edit]
set protocols rstp interface xe-0/0/6 bpdu-timeout-action block
```

Step-by-Step Procedure

To configure loop protection:

1. Configure interface **xe-0/0/6** on Switch 3:

```
[edit protocols rstp]
user@switch# set interface xe-0/0/6bpdu-timeout-action block
```

Results

Check the results of the configuration:

```
user@switch> show configuration protocols rstp
interface xe-0/0/6.0 {
  bpdu-timeout-action {
    block;
  }
}
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying the Interface State Before Loop Protection Is Triggered on page 183
- Verifying That Loop Protection Is Working on an Interface on page 183

Displaying the Interface State Before Loop Protection Is Triggered

Purpose Before loop protection is triggered on interface **xe-0/0/6**, confirm that the interface is blocked.

Action Display the interface state and role before applying root protection:

```
user@switch> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/0.0     128:513      128:513  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/1.0     128:514      128:514  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/2.0     128:515      128:515  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/3.0     128:516      128:516  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/4.0     128:517      128:517  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/5.0     128:518      128:518  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/6.0     128:519        128:2  16384.00aabbcc0348     20000  BLK    ALT
[output truncated]
```

Meaning The output from the operational mode command **show spanning-tree interface** shows that **xe-0/0/6.0** is the alternate port and is blocked.

Verifying That Loop Protection Is Working on an Interface

Purpose Verify that the loop protection configuration on interface **xe-0/0/6**. RSTP has been disabled on interface **xe-0/0/4** on Switch 1. This stops BPDUs from being sent to interface **xe-0/0/6** and triggering loop protection on that interface.

Action Display the interface state and role after applying root protection:

```
user@switch> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/0.0     128:513      128:513  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/1.0     128:514      128:514  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/2.0     128:515      128:515  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/3.0     128:516      128:516  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/4.0     128:517      128:517  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/5.0     128:518      128:518  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/6.0     128:519      128:519  32768.0019e2503f00     20000  BLK    DIS
(Loop-Incon)
[output truncated]
```

Meaning The operational mode command **show spanning-tree interface** shows that interface **xe-0/0/6.0** has detected that BPDUs are no longer being forwarded to it and has moved into a loop-inconsistent state. The loop-inconsistent state prevents the interface from

transitioning to a forwarding state. The interface recovers and transitions back to its original state as soon as it receives BPDUs.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Root Protection to Enforce Root Bridge Placement in Spanning Trees on page 207
- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Understanding Loop Protection for STP, RSTP, VSTP, and MSTP on page 56

Example: Configuring Network Regions for VLANs with MSTP

Multiple Spanning Tree Protocol (MSTP) is used to create a loop-free topology in networks using multiple spanning-tree regions, each region containing multiple spanning-tree instances (MSTIs). MSTIs provide different paths for different VLANs. This functionality facilitates more efficient load sharing across redundant links.

You can create up to 64 MSTI instances for QFX Series products, and each MSTI supports up to 4094 VLANs.

This example describes how to configure MSTP on four QFX3500 switches:

Requirements

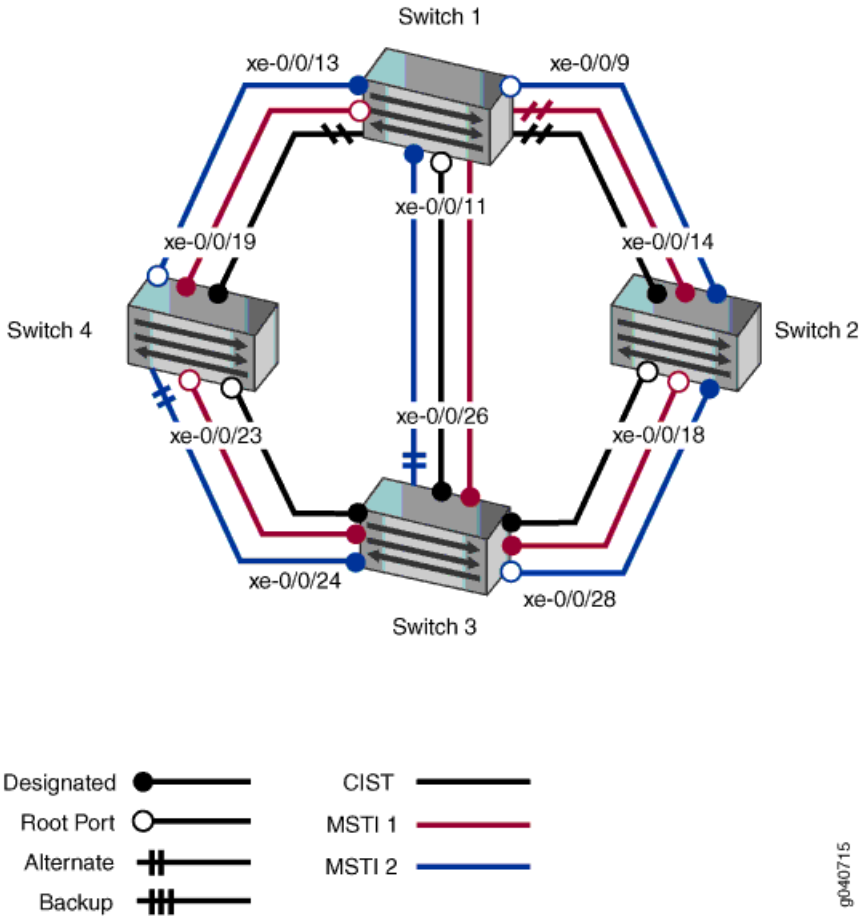
This example uses the following hardware and software components:

- Junos OS Release 11.1 for the QFX3500 switches
- Four QFX3500 switches

Overview and Topology

When the number of VLANs grows in a network, MSTP provides a more faster way of creating a loop-free topology using MSTIs. Each MSTI in the spanning-tree domain maintains its own tree. Each tree can be mapped to different links, utilizing bandwidth that would be unavailable to a single tree. MSTIs reduce demand on system resources.

Figure 17: Network Topology for MSTP



The interfaces shown in Table 26 on page 185 will be configured for MSTP.

NOTE: You can configure MSTP on logical or physical interfaces. This example shows MSTP configured on logical interfaces.

Table 26: Topology for Configuring MSTP on the QFX Series

Components	Settings
Switch 1	The following ports on Switch 1 are connected in this way: • **xe-0/0/9** is connected to Switch 2 • **xe-0/0/13** is connected to Switch 4 • **xe-0/0/11** is connected to Switch 3
Switch 2	The following ports on Switch 2 are connected in this way: • **xe-0/0/14** is connected to Switch 1 • **xe-0/0/18** is connected to Switch 3

Table 26: Topology for Configuring MSTP on the QFX Series *(continued)*

Components	Settings
Switch 3	The following ports on Switch 3 are connected in this way: • **xe-0/0/26** is connected to Switch 1 • **xe-0/0/28** is connected to Switch 2 • **xe-0/0/24** is connected to Switch 4
Switch 4	The following ports on Switch 4 are connected in this way: • **xe-0/0/19** is connected to Switch 1 • **xe-0/0/23** is connected to Switch 3
VLAN names and tag IDs	**sales-vlan**, tag **10** **engineering-vlan**, tag **20** **publications-vlan**, tag **30** **support-vlan**, tag **40**
MSTIs	**1** **2**

The topology in Figure 17 on page 185 shows a Common Internal Spanning Tree (CIST). The CIST is a single spanning tree connecting all devices in the network. The switch with the highest priority is elected as the root bridge of the CIST.

Also in an MSTP topology are ports that have specific roles:

- The *root port* is responsible for forwarding data to the root bridge.
- The *alternate port* is a standby port for the root port. When a root port goes down, the alternate port becomes the active root port.
- The *designated port* forwards data to the downstream network segment or device.
- The *backup port* is a backup port for the designated port. When a designated port goes down, the backup port becomes the active designated port and starts forwarding data.

In this example, one MSTP region, **region1**, contains Switch 1, Switch 2, Switch 3, and Switch 4. Within the region, four VLANs are created:

- The **sales-vlan** supports sales traffic and has a VLAN tag identifier of 10.
- The **engineering-vlan** supports data traffic and has a VLAN tag identifier of 20.
- The **publications-vlan** supports publications VLAN traffic (for supplicants that fail 802.1X authentication) and has a VLAN tag identifier of 30.
- The **support-vlan** supports video traffic and has a VLAN tag identifier of 40.

The VLANs are associated with specific interfaces on each of the four switches. Two MSTIs, **1** and **2**, are then associated with the VLAN tag identifiers, and some MSTP parameters, such as cost, are configured on each switch.

Configuring MSTP on Switch 1

CLI Quick Configuration

To quickly configure interfaces and MSTP on Switch 1, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/9 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/13 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/9 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
set protocols mstp configuration-name region1
set protocols mstp bridge-priority 16k
set protocols mstp interface xe-0/0/13.0 cost 1000
set protocols mstp interface xe-0/0/13.0 mode point-to-point
set protocols mstp interface xe-0/0/9.0 cost 1000
set protocols mstp interface xe-0/0/9.0 mode point-to-point
set protocols mstp interface xe-0/0/11.0 cost 1000
set protocols mstp interface xe-0/0/11.0 mode point-to-point
set protocols mstp msti 1 bridge-priority 16k
set protocols mstp msti 1 vlan [10 20]
set protocols mstp msti 1 interface xe-0/0/11.0 cost 4000
set protocols mstp msti 2 bridge-priority 8k
set protocols mstp msti 2 vlan [30 40]
```

Step-by-Step Procedure

To configure interfaces and MSTP on Switch 1:

1. Configure the VLANs **sales-vlan**, **engineering-vlan**, **publications-vlan**, and **support-vlan**:

   ```
   [edit vlans]
   user@switch1# set sales-vlan description “Sales VLAN”
   user@switch1# set sales-vlan vlan-id 10
   user@switch1# set engineering-vlan description “Engineering VLAN”
   user@switch1# set engineering-vlan vlan-id 20
   user@switch1# set publications-vlan description “Publications VLAN”
   user@switch1# set publications-vlan vlan-id 30
   user@switch1# set support-vlan description “Support VLAN”
   user@switch1# set publications-vlan vlan-id 40
   ```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch1# set xe-0/0/13 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch1# set xe-0/0/9 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch1# set xe-0/0/11 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the interface-mode statement instead of the port-mode statement. The port-mode statement has been replaced with the interface-mode statement.

```
[edit interfaces]
user@switch1# set xe-0/0/13 unit 0 family ethernet-switching port-mode trunk
user@switch1# set xe-0/0/9 unit 0 family ethernet-switching port-mode trunk
user@switch1# set xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
```

4. Configure MSTP on the switch, including the two MSTIs:

```
[edit protocols]
user@switch1# mstp configuration-name region1
user@switch1# mstp bridge-priority 16k
user@switch1# mstp interface xe-0/0/13.0 cost 1000
user@switch1# mstp interface xe-0/0/13.0 mode point-to-point
user@switch1# mstp interface xe-0/0/9.0 cost 1000
user@switch1# mstp interface xe-0/0/9.0 mode point-to-point
user@switch1# mstp interface xe-0/0/11.0 cost 4000
user@switch1# mstp interface xe-0/0/11.0 mode point-to-point
user@switch1# mstp msti 1 bridge-priority 16k
user@switch1# mstp msti 1 vlan [10 20]
user@switch1# mstp msti 1 interface xe-0/0/11.0 cost 4000
user@switch1# mstp msti 2 bridge-priority 8k
user@switch1# mstp msti 2 vlan [30 40]
```

Results Check the results of the configuration:

```
user@switch1> show configuration
interfaces {
    xe-0/0/13 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members 10;
                    members 20;
                    members 30;
                    members 40;
                }
            }
        }
    }
    xe-0/0/9 {
        unit 0 {
            family ethernet-switching {
```

```
            port-mode trunk;
            vlan {
              members 10;
              members 20;
              members 30;
              members 40;
            }
          }
        }
      }
      xe-0/0/11 {
        unit 0 {
          family ethernet-switching {
            port-mode trunk;
            vlan {
              members 10;
              members 20;
              members 30;
              members 40;
            }
          }
        }
      }
    }
    protocols {
      mstp {
        configuration-name region1;
      bridge-priority 16k;
        interface xe-0/0/13.0 {
          cost 1000;
          mode point-to-point;
        }
        interface xe-0/0/9.0 {
          cost 1000;
          mode point-to-point;
        }
        interface xe-0/0/11.0 {
          cost 4000;
          mode point-to-point;
        }
      msti 1 {
        bridge-priority 16k;
        vlan [ 10 20 ];
        interface xe-0/0/11.0 {
          cost 4000;
        }
      }
      msti 2 {
        bridge-priority 8k;
        vlan [ 30 40 ];
      }
    }
    vlans {
      sales-vlan {
        vlan-id 10;
      }
```

```
    engineering-vlan {
      vlan-id 20;
    }
    publications-vlan {
      vlan-id 30;
    }
    support-vlan {
      vlan-id 40;
    }
  }
```

Configuring MSTP on Switch 2

CLI Quick Configuration

To quickly configure interfaces and MSTP on Switch 2, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/18 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/18 unit 0 family ethernet-switching port-mode trunk
set protocols mstp configuration-name region1
set protocols mstp bridge-priority 32k
set protocols mstp interface xe-0/0/14.0 cost 1000
set protocols mstp interface xe-0/0/14.0 mode point-to-point
set protocols mstp interface xe-0/0/18.0 cost 1000
set protocols mstp interface xe-0/0/18.0 mode point-to-point
set protocols mstp msti 1 bridge-priority 32k
set protocols mstp msti 1 vlan [10 20]
set protocols mstp msti 2 bridge-priority 4k
set protocols mstp msti 2 vlan [30 40]
```

Step-by-Step Procedure

To configure interfaces and MSTP on Switch 2:

1. Configure the VLANs **sales-vlan**, **engineering-vlan**, **publications-vlan**, and **support-vlan**:

```
[edit vlans]
user@switch2# set sales-vlan description “Sales VLAN”
user@switch2# set sales-vlan vlan-id 10
user@switch2# set engineering-vlan description “Engineering VLAN”
user@switch2# set engineering-vlan vlan-id 20
user@switch2# set publications-vlan description “Publications VLAN”
user@switch2# set publications-vlan vlan-id 30
user@switch2# set support-vlan vlan-description “Support VLAN”
user@switch2# set publications-vlan vlan-id 40
```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch2# set xe-0/0/14 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch2# set xe-0/0/18 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit interfaces]
user@switch2# set xe-0/0/14 unit 0 family ethernet-switching port-mode trunk
user@switch2# set xe-0/0/18 unit 0 family ethernet-switching port-mode trunk
```

4. Configure MSTP on the switch, including the two MSTIs:

```
[edit protocols]
user@switch2# mstp configuration-name region1
user@switch2# mstp bridge-priority 32k
user@switch2# mstp interface xe-0/0/14.0 cost 1000
user@switch2# mstp interface xe-0/0/14.0 mode point-to-point
user@switch2# mstp interface xe-0/0/18.0 cost 1000
user@switch2# mstp interface xe-0/0/18.0 mode point-to-point
user@switch2# mstp interface all cost 1000
user@switch2# mstp msti 1 bridge-priority 32k
user@switch2# mstp msti 1 vlan [10 20]
user@switch2# mstp msti 2 bridge-priority 4k
user@switch2# mstp msti 2 vlan [30 40]
```

Results

Check the results of the configuration:

```
user@switch2> show configuration
interfaces {
   xe-0/0/14 {
      unit 0 {
         family ethernet-switching {
            port-mode trunk;
            vlan {
```

```
                members 10;
                members 20;
                members 30;
                members 40;
            }
          }
        }
      }
      xe-0/0/18 {
        unit 0 {
          family ethernet-switching {
            port-mode trunk;
            vlan {
              members 10;
              members 20;
              members 30;
              members 40;
            }
          }
        }
      }
    }
    protocols {
      mstp {
        configuration-name region1;
        bridge-priority 32k;
        interface xe-0/0/14.0 {
          cost 1000;
          mode point-to-point;
        }
        interface xe-0/0/18.0 {
          cost 1000;
          mode point-to-point;
        }
        msti 1 {
          bridge-priority 32k;
          vlan [ 10 20 ];
        }
        msti 2 {
          bridge-priority 4k;
          vlan [ 30 40 ];
        }
      }
    }
    vlans {
      sales-vlan {
        vlan-id 10;
      }
      engineering-vlan {
        vlan-id 20;
      }
      publications-vlan {
        vlan-id 30;
      }
      support-vlan {
        vlan-id 40;
```

```
    }
}
```

Configuring MSTP on Switch 3

CLI Quick Configuration

To quickly configure interfaces and MSTP on Switch 3, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/26 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/28 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/24 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/26 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/28 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/24 unit 0 family ethernet-switching port-mode trunk
set protocols mstp configuration-name region1
set protocols mstp bridge-priority 8k
set protocols mstp interface xe-0/0/26.0 cost 1000
set protocols mstp interface xe-0/0/26.0 mode point-to-point
set protocols mstp interface xe-0/0/28.0 cost 1000
set protocols mstp interface xe-0/0/28.0 mode point-to-point
set protocols mstp interface xe-0/0/24.0 cost 1000
set protocols mstp interface xe-0/0/24.0 mode point-to-point
set protocols mstp msti 1 bridge-priority 4k
set protocols mstp msti 1 vlan [10 20]
set protocols mstp msti 2 bridge-priority 16k
set protocols mstp msti 2 vlan [30 40]
```

Step-by-Step Procedure

To configure interfaces and MSTP on Switch 3:

1. Configure the VLANs **sales-vlan**, **engineering-vlan**, **publications-vlan**, and **support-vlan**:

```
[edit vlans]
user@switch3# set sales-vlan description “Sales VLAN”
user@switch3# set sales-vlan vlan-id 10
user@switch3# set engineering-vlan description “Engineering VLAN”
user@switch3# set engineering-vlan vlan-id 20
user@switch3# set publications-vlan description “Publications VLAN”
user@switch3# set publications-vlan vlan-id 30
user@switch3# set support-vlan description “Support VLAN”
user@switch3# set publications-vlan vlan-id 40
```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch3# set xe-0/0/26 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch3# set xe-0/0/28 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch3# set xe-0/0/24 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the interface-mode statement instead of the port-mode statement. The port-mode statement has been replaced with the interface-mode statement.

```
[edit interfaces]
user@switch3# set xe-0/0/26 unit 0 family ethernet-switching port-mode trunk
user@switch3# set xe-0/0/28 unit 0 family ethernet-switching port-mode trunk
user@switch3# set xe-0/0/24 unit 0 family ethernet-switching port-mode trunk
```

4. Configure MSTP on the switch, including the two MSTIs:

```
[edit protocols]
user@switch3# mstp configuration-name region1
user@switch3# mstp bridge-priority 8k
user@switch3# mstp interface xe-0/0/26.0 cost 1000
user@switch3# mstp interface xe-0/0/26.0 mode point-to-point
user@switch3# mstp interface xe-0/0/28.0 cost 1000
user@switch3# mstp interface xe-0/0/28.0 mode point-to-point
user@switch3# mstp interface xe-0/0/24.0 cost 1000
user@switch3# mstp interface xe-0/0/24.0 mode point-to-point
user@switch3# mstp interface all cost 1000
user@switch3# mstp msti 1 bridge-priority 4k
user@switch3# mstp msti 1 vlan [10 20]
user@switch3# mstp msti 2 bridge-priority 16k
user@switch3# mstp msti 2 vlan [30 40]
```

Results Check the results of the configuration:

```
user@switch3> show configuration
  interfaces {
    xe-0/0/26 {
      unit 0 {
        family ethernet-switching {
          port-mode trunk;
          vlan {
            members 10;
            members 20;
            members 30;
            members 40;
          }
        }
      }
    }
```

```
    xe-0/0/28 {
      unit 0 {
        family ethernet-switching {
          port-mode trunk;
          vlan {
            members 10;
            members 20;
            members 30;
            members 40;
          }
        }
      }
    }
    xe-0/0/24 {
      unit 0 {
        family ethernet-switching {
          port-mode trunk;
          vlan {
            members 10;
            members 20;
            members 30;
            members 40;
          }
        }
      }
    }
  }
}
protocols {
  mstp {
    configuration-name region1;
    bridge-priority 8k;
    interface xe-0/0/26.0 {
      cost 1000;
      mode point-to-point;
    }
    interface xe-0/0/28.0 {
      cost 1000;
      mode point-to-point;
    }
    interface xe-0/0/24.0 {
      cost 1000;
      mode point-to-point;
    }
    msti 1 {
      bridge-priority 4k;
      vlan [ 10 20 ];
    }
    msti 2 {
      bridge-priority 16k;
      vlan [ 30 40 ];
    }
  }
}
vlans {
  sales-vlan {
```

```
    vlan-id 10;
  }
  engineering-vlan {
    vlan-id 20;
  }
  publications-vlan {
    vlan-id 30;
  }
  support-vlan {
    vlan-id 40;
  }
}
```

Configuring MSTP on Switch 4

CLI Quick Configuration

To quickly configure interfaces and MSTP on Switch 4, copy the following commands and paste them into the switch terminal window:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the interface-mode statement instead of the port-mode statement. The port-mode statement has been replaced with the interface-mode statement.

```
[edit]
set vlans sales-vlan description “Sales VLAN”
set vlans sales-vlan vlan-id 10
set vlans engineering-vlan description “Engineering VLAN”
set vlans engineering-vlan vlan-id 20
set vlans publications-vlan description “Publications VLAN”
set vlans publications-vlan vlan-id 30
set vlans support-vlan description “Support VLAN”
set vlans support-vlan vlan-id 40
set interfaces xe-0/0/23 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/19 unit 0 family ethernet-switching vlan members [10 20 30 40]
set interfaces xe-0/0/23 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/19 unit 0 family ethernet-switching port-mode trunk
set protocols mstp configuration-name region1
set protocols mstp bridge-priority 16k
set protocols mstp interface xe-0/0/23.0 cost 1000
set protocols mstp interface xe-0/0/23.0 mode point-to-point
set protocols mstp interface xe-0/0/19.0 cost 1000
set protocols mstp interface xe-0/0/19.0 mode point-to-point
set protocols mstp msti 1 bridge-priority 16k
set protocols mstp msti 1 vlan [10 20]
set protocols mstp msti 2 bridge-priority 32k
set protocols mstp msti 2 vlan [30 40]
```

Step-by-Step Procedure

To configure interfaces and MSTP on Switch 4:

1. Configure the VLANs **sales-vlan**, **engineering-vlan**, **publications-vlan**, and **support-vlan**:

```
[edit vlans]
user@switch4# set sales-vlan description “Sales VLAN”
user@switch4# set sales-vlan vlan-id 10
user@switch4# set engineering-vlan description “Engineering VLAN”
user@switch4# set engineering-vlan vlan-id 20
user@switch4# set publications-vlan description “Publications VLAN”
user@switch4# set publications-vlan vlan-id 30
user@switch4# set support-vlan description “Support VLAN”
user@switch4# set publications-vlan vlan-id 40
```

2. Configure the VLANs on the interfaces, including support for the Ethernet switching protocol:

```
[edit interfaces]
user@switch4# set xe-0/0/23 unit 0 family ethernet-switching vlan members [10 20 30
40]
user@switch4# set xe-0/0/19 unit 0 family ethernet-switching vlan members [10 20 30
40]
```

3. Configure the port mode for the interfaces:

NOTE: If you are configuring MSTP on devices that support the Enhanced Layer 2 Switching (ELS) CLI, use the **interface-mode** statement instead of the **port-mode** statement. The **port-mode** statement has been replaced with the **interface-mode** statement.

```
[edit interfaces]
user@switch4# set ge-0/0/23 unit 0 family ethernet-switching port-mode trunk
user@switch4# set ge-0/0/19 unit 0 family ethernet-switching port-mode trunk
```

4. Configure MSTP on the switch, including the two MSTIs:

```
[edit protocols]
user@switch4# mstp configuration-name region1
user@switch4# mstp bridge-priority 16k
user@switch4# mstp interface all cost 1000
user@switch4# mstp interface xe-0/0/23.0 cost 1000
user@switch4# mstp interface xe-0/0/23.0 mode point-to-point
user@switch4# mstp interface xe-0/0/19.0 cost 1000
user@switch4# mstp interface xe-0/0/19.0 mode point-to-point
user@switch4# mstp msti 1 bridge-priority 16k
user@switch4# mstp msti 1 vlan [10 20]
user@switch4# mstp msti 2 bridge-priority 32k
user@switch4# mstp msti 2 vlan [30 40]
```

Results

Check the results of the configuration:

```
user@switch4> show configuration
interfaces {
   xe-0/0/23 {
      unit 0 {
         family ethernet-switching {
            port-mode trunk;
            vlan {
```

```
                members 10;
                members 20;
                members 30;
                members 40;
            }
          }
        }
      }
      xe-0/0/19 {
        unit 0 {
          family ethernet-switching {
            port-mode trunk;
            vlan {
              members 10;
              members 20;
              members 30;
              members 40;
            }
          }
        }
      }
    }
    protocols {
      mstp {
        configuration-name region1;
        bridge-priority 16k;
        interface xe-0/0/23.0 {
          cost 1000;
          mode point-to-point;
        }
        interface xe-0/0/19.0 {
          cost 1000;
          mode point-to-point;
        }
        msti 1 {
          bridge-priority 16k;
          vlan [ 10 20 ];
        }
        msti 2 {
          bridge-priority 32k;
          vlan [ 30 40 ];
        }
      }
    }
    vlans {
      sales-vlan {
        vlan-id 10;
      }
      engineering-vlan {
        vlan-id 20;
      }
      publications-vlan {
        vlan-id 30;
      }
      support-vlan {
        vlan-id 40;
```

```
    }
}
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying MSTP Configuration on Switch 1 on page 199
- Verifying MSTP Configuration on Switch 2 on page 201
- Verifying MSTP Configuration on Switch 3 on page 203
- Verifying MSTP Configuration on Switch 4 on page 205

Verifying MSTP Configuration on Switch 1

Purpose Verify the MSTP configuration on Switch 1.

Action Use the operational mode commands:

```
user@switch1> show spanning-tree interface
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/13.0    128:527      128:525  16384.0019e25040e0      1000  FWD    ROOT
xe-0/0/9.0     128:529      128:513  32768.0019e2503d20      1000  BLK    ALT
xe-0/0/11.0    128:531      128:513   8192.0019e25051e0      4000  BLK    ALT

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/13.0    128:527      128:525  16385.0019e25040e0      1000  FWD    ROOT
xe-0/0/9.0     128:529      128:513  32769.0019e2503d20      1000  BLK    ALT
xe-0/0/11.0    128:531      128:513   4097.0019e25051e0      4000  BLK    ALT

Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/13.0    128:527      128:527   8194.0019e25044e0      1000  FWD    DESG
xe-0/0/9.0     128:529      128:513   4098.0019e2503d20      1000  FWD    ROOT
xe-0/0/11.0    128:531      128:531   8194.0019e25044e0      1000  FWD    DESG


user@switch1> show spanning-tree bridge
STP bridge parameters
Context ID                          : 0
Enabled protocol                    : MSTP

STP bridge parameters for CIST
  Root ID                           : 8192.00:19:e2:50:51:e0
  Root cost                         : 0
  Root port                         : xe-0/0/13.0
  CIST regional root                : 8192.00:19:e2:50:51:e0
  CIST internal root cost           : 2000
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Message age                       : 0
  Number of topology changes        : 3
  Time since last topology change   : 921 seconds
  Local parameters
    Bridge ID                       : 16384.00:19:e2:50:44:e0
    Extended system ID              : 0
    Internal instance ID            : 0

STP bridge parameters for MSTI 1
  MSTI regional root                : 4097.00:19:e2:50:51:e0
  Root cost                         : 2000
  Root port                         : xe-0/0/13.0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Local parameters
    Bridge ID                       : 16385.00:19:e2:50:44:e0
```

```
    Extended system ID              : 0
    Internal instance ID            : 1

STP bridge parameters for MSTI 2
  MSTI regional root                : 4098.00:19:e2:50:3d:20
  Root cost                         : 1000
  Root port                         : xe-0/0/9.0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 19
  Local parameters
    Bridge ID                       : 8194.00:19:e2:50:44:e0
    Extended system ID              : 0
    Internal instance ID            : 2
```

Meaning The operational mode command **show spanning-tree interface** displays spanning-tree domain information such as the designated port and the port roles.

The operational mode command **show spanning-tree bridge** displays the spanning-tree domain information at either the bridge level or the interface level. If the optional interface name is omitted, all interfaces in the spanning-tree domain are displayed.

Verifying MSTP Configuration on Switch 2

Purpose Verify the MSTP configuration on Switch 2.

Action Use the operational mode commands:

```
user@switch2> show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/14.0     128:513      128:513  32768.0019e2503d20      1000  FWD    DESG
xe-0/0/18.0     128:519      128:515   8192.0019e25051e0      1000  FWD    ROOT

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/14.0     128:513      128:513  32769.0019e2503d20      1000  FWD    DESG
xe-0/0/18.0     128:519      128:515   4097.0019e25051e0      1000  FWD    ROOT

Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/14.0     128:513      128:513   4098.0019e2503d20      1000  FWD    DESG

xe-0/0/18.0     128:519      128:519   4098.0019e2503d20      1000  FWD    DESG

user@switch2> show spanning-tree bridge
STP bridge parameters
Context ID                          : 0
Enabled protocol                    : MSTP

STP bridge parameters for CIST
  Root ID                           : 8192.00:19:e2:50:51:e0
  Root cost                         : 0
  Root port                         : xe-0/0/18.0
  CIST regional root                : 8192.00:19:e2:50:51:e0
  CIST internal root cost           : 1000
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 19
  Message age                       : 0
  Number of topology changes        : 1
  Time since last topology change   : 782 seconds
  Local parameters
    Bridge ID                       : 32768.00:19:e2:50:3d:20
    Extended system ID              : 0
    Internal instance ID            : 0

STP bridge parameters for MSTI 1
  MSTI regional root                : 4097.00:19:e2:50:51:e0
  Root cost                         : 1000
  Root port                         : xe-0/0/18.0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 19
  Local parameters
    Bridge ID                       : 32769.00:19:e2:50:3d:20
```



```
    Extended system ID              : 0
    Internal instance ID            : 1

STP bridge parameters for MSTI 2
  MSTI regional root                : 4098.00:19:e2:50:3d:20
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Local parameters
    Bridge ID                       : 4098.00:19:e2:50:3d:20
    Extended system ID              : 0
    Internal instance ID            : 2
```

Meaning The operational mode command **show spanning-tree interface** displays spanning-tree domain information such as the designated port and the port roles.

The operational mode command **show spanning-tree bridge** displays the spanning-tree domain information at either the bridge level or the interface level. If the optional interface name is omitted, all interfaces in the spanning-tree domain are displayed.

Verifying MSTP Configuration on Switch 3

Purpose Verify the MSTP configuration on Switch 3.

Action Use the operational mode commands:

```
user@switch3>  show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/26.0     128:513      128:513   8192.0019e25051e0      1000  FWD    DESG
xe-0/0/28.0     128:515      128:515   8192.0019e25051e0      1000  FWD    DESG
xe-0/0/24.0     128:517      128:517   8192.0019e25051e0      1000  FWD    DESG

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/26.0     128:513      128:513   4097.0019e25051e0      1000  FWD    DESG
xe-0/0/28.0     128:515      128:515   4097.0019e25051e0      1000  FWD    DESG
xe-0/0/24.0     128:517      128:517   4097.0019e25051e0      1000  FWD    DESG


Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/26.0     128:513      128:531   8194.0019e25044e0      1000  BLK    ALT
xe-0/0/28.0     128:515      128:519   4098.0019e2503d20      1000  FWD    ROOT
xe-0/0/24.0     128:517      128:517  16386.0019e25051e0      1000  FWD    DESG


user@switch3>  show spanning-tree bridge
STP bridge parameters
Context ID                          : 0
Enabled protocol                    : MSTP

STP bridge parameters for CIST
  Root ID                           : 8192.00:19:e2:50:51:e0
  CIST regional root                : 8192.00:19:e2:50:51:e0
  CIST internal root cost           : 0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Number of topology changes        : 3
  Time since last topology change   : 843 seconds
  Local parameters
    Bridge ID                       : 8192.00:19:e2:50:51:e0
    Extended system ID              : 0
    Internal instance ID            : 0

STP bridge parameters for MSTI 1
  MSTI regional root                : 4097.00:19:e2:50:51:e0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Local parameters
    Bridge ID                       : 4097.00:19:e2:50:51:e0
    Extended system ID              : 0
    Internal instance ID            : 1

STP bridge parameters for MSTI 2
  MSTI regional root                : 4098.00:19:e2:50:3d:20
```

```
Root cost                       : 1000
Root port                       : xe-0/0/28.0
Hello time                      : 2 seconds
Maximum age                     : 20 seconds
Forward delay                   : 15 seconds
Hop count                       : 19
Local parameters
  Bridge ID                     : 16386.00:19:e2:50:51:e0
  Extended system ID            : 0
  Internal instance ID          : 2
```

Meaning The operational mode command **show spanning-tree interface** displays spanning-tree domain information such as the designated port and the port roles.

The operational mode command **show spanning-tree bridge** displays the spanning-tree domain information at either the bridge level or the interface level. If the optional interface name is omitted, all interfaces in the spanning-tree domain are displayed.

Verifying MSTP Configuration on Switch 4

Purpose Verify the MSTP configuration on Switch 4.

Action Use the operational mode commands:

```
user@switch4> show spanning-tree interface
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost

xe-0/0/23.0    128:523      128:517   8192.0019e25051e0      1000  FWD    ROOT
xe-0/0/19.0    128:525      128:525  16384.0019e25040e0      1000  FWD    DESG

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/23.0    128:523      128:517   4097.0019e25051e0      1000  FWD    ROOT
xe-0/0/19.0    128:525      128:525  16385.0019e25040e0      1000  FWD    DESG


Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/23.0    128:523      128:517  16386.0019e25051e0      1000  BLK    ALT
xe-0/0/19.0    128:525      128:527   8194.0019e25044e0      1000  FWD    ROOT


user@switch4> show spanning-tree bridge
STP bridge parameters
Context ID                          : 0
Enabled protocol                    : MSTP

STP bridge parameters for CIST
  Root ID                           : 8192.00:19:e2:50:51:e0
  Root cost                         : 0
  Root port                         : xe-0/0/23.0
  CIST regional root                : 8192.00:19:e2:50:51:e0
  CIST internal root cost           : 1000
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 19
  Message age                       : 0
  Number of topology changes        : 4
  Time since last topology change   : 887 seconds
  Local parameters
    Bridge ID                       : 16384.00:19:e2:50:40:e0
    Extended system ID              : 0
    Internal instance ID            : 0

STP bridge parameters for MSTI 1
  MSTI regional root                : 4097.00:19:e2:50:51:e0
  Root cost                         : 1000
  Root port                         : xe-0/0/23.0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 19
  Local parameters
    Bridge ID                       : 16385.00:19:e2:50:40:e0
    Extended system ID              : 0
```

```
    Internal instance ID            : 1

STP bridge parameters for MSTI 2
  MSTI regional root                : 4098.00:19:e2:50:3d:20
  Root cost                         : 2000
  Root port                         : xe-0/0/19.0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Local parameters
    Bridge ID                       : 32770.00:19:e2:50:40:e0
    Extended system ID              : 0
    Internal instance ID            : 2
```

Meaning The operational mode command **show spanning-tree interface** displays spanning-tree domain information such as the designated port and the port roles.

The operational mode command **show spanning-tree bridge** displays the spanning-tree domain information at either the bridge level or the interface level. If the optional interface name is omitted, all interfaces in the spanning-tree domain are displayed.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52

Example: Configuring Root Protection to Enforce Root Bridge Placement in Spanning Trees

QFX Series products provide Layer 2 loop prevention through Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP). Root protection increases the efficiency of STP, RSTP, and MSTP by allowing network administrators to enforce the root bridge placement in the network manually.

This example describes how to configure root protection on an interface for the QFX Series.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 11.1 or later for the QFX Series
- Four switches in an RSTP topology

Before you configure the interface for root protection, be sure you have:

- RSTP operating on the switches.

NOTE: By default, RSTP is enabled on the QFX Series.

Overview and Topology

Peer STP applications running on switch interfaces exchange a special type of frame called a bridge protocol data unit (BPDU). Switches communicate interface information using BPDUs to create a loop-free topology that ultimately determines the root bridge and which interfaces block or forward traffic in the spanning tree.

You can also see BPDUs generated when you run a bridge application on a device attached to the switch. This can interfere with root port election, which may sometimes lead to the wrong root port being elected through the above process. Root protection allows you to manually enforce the root bridge placement in the network.

To prevent this from happening, enable root protection on interfaces that should not receive more BPDUs from the root bridge and should not be elected as the root port. These interfaces are typically located on an administrative boundary and are designated ports.

When root protection is enabled on an interface:

- The interface is blocked from becoming the root port.
- Root protection is enabled for all STP instances on that interface.
- The interface is blocked only for instances for which it receives more BPDUs. Otherwise, it participates in the spanning-tree topology.

NOTE: An interface can be configured for either root protection or loop protection, but not for both.

Four switches are displayed in Figure 18 on page 209. In this example, they are configured for RSTP and create a loop-free topology. Interface **xe-0/0/7** on Switch 1 is a designated port on an administrative boundary. It connects to Switch 4. Switch 3 is the root bridge. Interface **xe-0/0/6** on Switch 1 is the root port.

This example shows how to configure root protection on interface **xe-0/0/7** to prevent it from transitioning to become the root port.

Figure 18: Network Topology for Root Protection

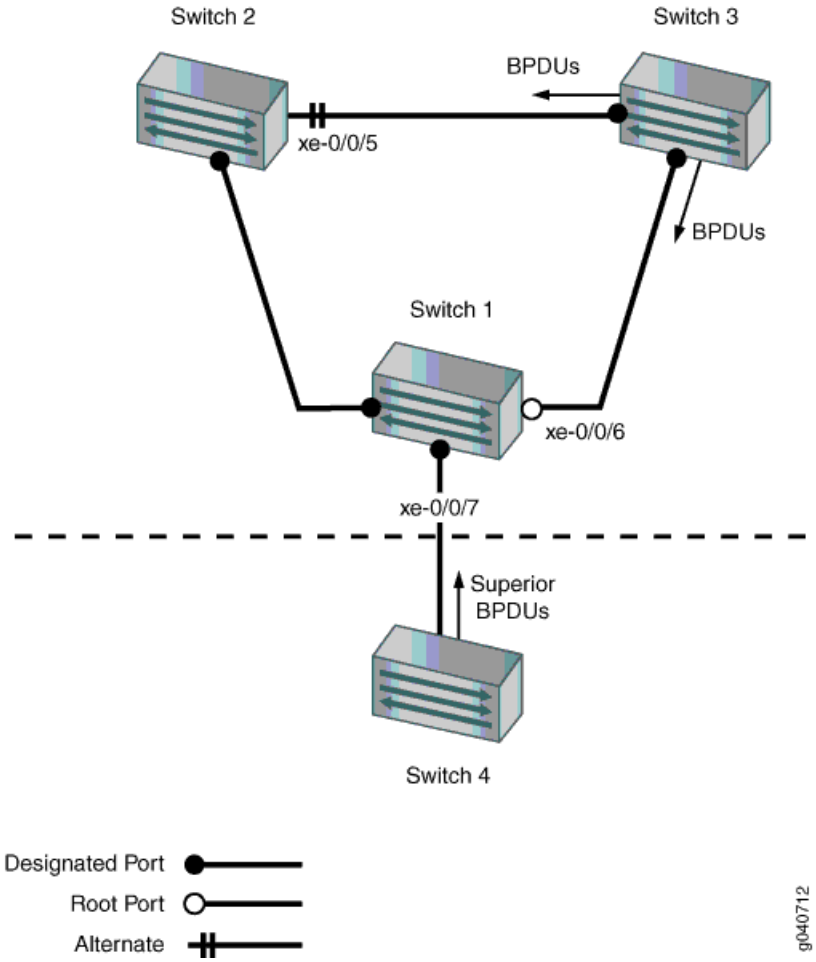


Table 27 on page 209 shows the components that will be configured for root protection.

Table 27: Topology for Configuring Root Protection on the QFX Series

Component	Settings
Switch 1	Switch 1 is connected to Switch 4 through interface **xe-0/0/7**.
Switch 2	Switch 2 is connected to Switch 1 and Switch 3. Interface **xe-0/0/4** is the alternate port in the RSTP topology.
Switch 3	Switch 3 is the root bridge and is connected to Switch 1 and Switch 2.
Switch 4	Switch 4 is connected to Switch 1. After loop protection is configured on interface **xe-0/0/7**, Switch 4 sends more BPDUs that trigger loop protection on interface **xe-0/0/7**.

A spanning-tree topology contains ports that have specific roles:

- The *root port* is responsible for forwarding data to the root bridge.

- The *alternate port* is a standby port for the root port. When a root port goes down, the alternate port becomes the active root port.
- The *designated port* forwards data to the downstream network segment or device.

This configuration example uses an RSTP topology. However, you can also configure root protection for STP or MSTP topologies at the [**edit protocols** (**mstp** | **stp**)] hierarchy level.

Configuration

CLI Quick Configuration

To quickly configure root protection on interface **xe-0/0/7**, copy the following command and paste it into the switch terminal window:

```
[edit]
set protocols rstp interface xe-0/0/7 no-root-port
```

Step-by-Step Procedure

To configure root protection:

1. Configure interface **xe-0/0/7**:

```
[edit protocols rstp]
user@switch#
set interface xe-0/0/7 no-root-port
```

Results

Check the results of the configuration:

```
user@switch> show configuration protocols rstp
interface xe-0/0/7.0 {
  no-root-port;
}
```

Verification

To confirm that the configuration is working properly, perform these tasks:

- Displaying the Interface State Before Root Protection Is Triggered on page 210
- Verifying That Root Protection Is Working on the Interface on page 211

Displaying the Interface State Before Root Protection Is Triggered

Purpose

Before root protection is triggered on interface **xe-0/0/7**, confirm the interface state.

Action Confirm the state of the interfaces before root protection is configured:

```
user@switch>  show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/0.0     128:513      128:513  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/1.0     128:514      128:514  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/2.0     128:515      128:515  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/3.0     128:516      128:516  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/4.0     128:517      128:517  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/5.0     128:518        128:2  16384.00aabbcc0348     20000  BLK    ALT
xe-0/0/6.0     128:519        128:1  16384.00aabbcc0348     20000  FWD    ROOT
xe-0/0/7.0     128:520      128:520  32768.0019e2503f00     20000  FWD    DESG
[output truncated]
```

Meaning The output from the operational mode command **show spanning-tree interface** shows that **xe-0/0/7.0** is a designated port in a forwarding state.

Verifying That Root Protection Is Working on the Interface

Purpose A configuration change takes place on Switch 4. A lower bridge priority on Switch 4 causes it to send more BPDUs to interface **xe-0/0/7**. Receipt of more BPDUs on interface **xe-0/0/7** triggers root protection. Verify that root protection is operating on interface **xe-0/0/7**.

Action Verify that root protection has been configured and is operating correctly:

```
user@switch>  show spanning-tree interface

Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-0/0/0.0     128:513      128:513  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/1.0     128:514      128:514  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/2.0     128:515      128:515  32768.0019e2503f00     20000  BLK    DIS
xe-0/0/3.0     128:516      128:516  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/4.0     128:517      128:517  32768.0019e2503f00     20000  FWD    DESG
xe-0/0/5.0     128:518        128:2  16384.00aabbcc0348     20000  BLK    ALT
xe-0/0/6.0     128:519        128:1  16384.00aabbcc0348     20000  FWD    ROOT
xe-0/0/7.0     128:520      128:520  32768.0019e2503f00     20000  BLK    DIS
(Root-Incon)
[output truncated]
```

Meaning The operational mode command show spanning-tree interface shows that interface **xe-0/0/7.0** has transitioned to a loop inconsistent state. The loop inconsistent state blocks the interface and prevents it from becoming a candidate for the root port. When the root bridge no longer receives more STP BPDUs from the interface, the interface recovers and transitions back to a forwarding state. Recovery is automatic.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Understanding Root Protection for STP, RSTP, VSTP, and MSTP on page 57

Example: Configuring Routing Between VLANs on One Switch

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs). For example, you might want to create a VLAN that includes the employees in a department and the resources that they use often, such as printers, servers, and so on.

Of course, you also want to allow these employees to communicate with people and resources in other VLANs. To forward packets between VLANs you normally you need a router that connects the VLANs. However, you can accomplish this on a Juniper Networks switch without using a router by configuring an integrated routing and bridging (IRB) interface (also known as a routed VLAN interface—or RVI—in versions of Junos OS that do not support Enhanced Layer 2 Software). Using this approach reduces complexity and avoids the costs associated with purchasing, installing, managing, powering, and cooling another device.

Requirements

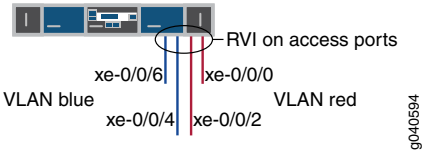
This example uses the following hardware and software components:

- One switch
- Junos OS Release 11.1 or later

Overview and Topology

This example uses an IRB to route traffic between two VLANs on the same switch. The topology is shown in Figure 19 on page 212.

Figure 19: IRB with One Switch



This example shows a simple configuration to illustrate the basic steps for creating two VLANs on a single switch and configuring an IRB to enable routing between the VLANs. One VLAN, called **blue**, is for the sales and marketing group, and a second, called **red**, is for the customer support team. The sales and support groups each have their own file servers and wireless access points. Each VLAN must have a unique name, tag (VLAN ID), and distinct IP subnet. Table 28 on page 213 lists the components of the sample topology.

Table 28: Components of the Multiple VLAN Topology

Property	Settings
VLAN names and tag IDs	**blue**, ID **100** **red**, ID **200**
Subnets associated with VLANs	**blue**: **192.0.2.0/25** (addresses **192.0.2.1** through **192.0.2.126**) **red**: **192.0.2.128/25** (addresses **192.0.2.129** through **192.0.2.254**)
Interfaces in VLAN **blue**	Sales server port: **xe-0/0/4** Sales wireless access points: **xe-0/0/6**
Interfaces in VLAN **red**	Support server port: **xe-0/0/0** Support wireless access points: **xe-0/0/2**
IRB name	interface **irb**
IRB units and addresses	logical unit 100: **192.0.2.1/25** logical unit 200: **192.0.2.129/25**

This configuration example creates two IP subnets, one for the blue VLAN and the second for the red VLAN. The switch bridges traffic within the VLANs. For traffic passing between two VLANs, the switch routes the traffic using an IRB on which you have configured addresses in each IP subnet.

To keep the example simple, the configuration steps show only a few interfaces and VLANs. Use the same configuration procedure to add more interfaces and VLANs. By default, all interfaces are in access mode, so you do not have to configure the port mode.

Configure Layer 2 switching for two VLANs

CLI Quick Configuration

To quickly configure Layer 2 switching for the two VLANs (**blue** and **red**) and to quickly configure Layer 3 routing of traffic between the two VLANs, copy the following commands and paste them into the switch terminal window:

NOTE: The following example uses a version of Junos OS that supports Enhanced Layer 2 Software (ELS). When you use ELS, you create a Layer 3 virtual interface named *irb*. If you are using a version of Junos OS that does not support ELS, you create a Layer 3 virtual interface named vlan.

```
[edit]
set interfaces xe-0/0/4 unit 0 description “Sales server port”
```

```
set interfaces xe-0/0/4 unit 0 family ethernet-switching vlan members blue
set interfaces xe-0/0/6 unit 0 description “Sales wireless access point port”
set interfaces xe-0/0/6 unit 0 family ethernet-switching vlan members blue
set interfaces xe-0/0/0 unit 0 description “Support servers”
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members red
set interfaces xe-0/0/2 unit 0 description “Support wireless access point port”
set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members red
set interfaces irb unit 100 family inet address 192.0.2.1/25
set interfaces irb unit 200 family inet address 192.0.2.129/25
set vlans blue l3-interface irb.100
set vlans blue vlan-id 100
set vlans red vlan-id 200
set vlans red l3-interface irb.200
```

Step-by-Step Procedure

To configure the switch interfaces and the VLANs to which they belong:

1. Configure the interface for the sales server in the blue VLAN:

   ```
   [edit interfaces xe-0/0/4 unit 0]
   user@switch# set description “Sales server port”
   user@switch# set family ethernet-switching vlan members blue
   ```

2. Configure the interface for the wireless access point in the blue VLAN:

   ```
   [edit interfaces xe-0/0/6 unit 0]
   user@switch# set description “Sales wireless access point port”
   user@switch# set family ethernet-switching vlan members blue
   ```

3. Configure the interface for the support server in the red VLAN:

   ```
   [edit interfaces xe-0/0/0 unit 0]
   user@switch# set description “Support server port”
   user@switch# set family ethernet-switching vlan members red
   ```

4. Configure the interface for the wireless access point in the red VLAN:

   ```
   [edit interfaces xe-0/0/2 unit 0]
   user@switch# set description “Support wireless access point port”
   user@switch# set family ethernet-switching vlan members red
   ```

Step-by-Step Procedure

Now create the VLANs and the IRB. The IRB will have logical units in the broadcast domains of both VLANs.

1. Create the red and blue VLANs by configuring the VLAN IDs for them:

   ```
   [edit vlans]
   user@switch# set blue vlan-id 100
   user@switch# set red vlan-id 200
   ```

2. Create the interface named **irb** with a logical unit in the sales broadcast domain (blue VLAN):

   ```
   [edit interfaces]
   user@switch# set irb unit 100 family inet address 192.0.2.1/25
   ```

   The unit number is arbitrary and does not have to match the VLAN tag ID. However, configuring the unit number to match the VLAN ID can help avoid confusion.

3. Add a logical unit in the support broadcast domain (red VLAN) to the **irb** interface:

   ```
   [edit interfaces]
   user@switch# set irb unit 200 family inet address 192.0.2.129/25
   ```

4. Complete the IRB configuration by binding the red and blue VLANs (Layer 2) with the appropriate logical units of the **irb** interface (Layer 3):

```
[edit vlans]
user@switch# set blue l3-interface irb.100
user@switch# set red l3-interface irb.200
```

Display the results of the configuration:

```
user@switch> show configuration
interfaces {
    xe-0/0/4 {
        unit 0 {
            description “Sales server port”;
            family ethernet-switching {
                vlan members blue;
            }
        }
    }
    xe-0/0/6 {
        unit 0 {
            description “Sales wireless access point port”;
            family ethernet-switching {
                vlan members blue;
            }
        }
    }
    xe-0/0/0 {
        unit 0 {
            description “Support server port”;
            family ethernet-switching {
                vlan members red;
            }
        }
    }
    xe-0/0/2 {
        unit 0 {
            description “Support wireless access point port”;
            family ethernet-switching {
                vlan members red;
            }
        }
    }
    irb {
        unit 100 {
            family inet address 192.0.2.1/25;
        }
        unit 200 {
            family inet address 192.0.2.129/25;
            }
        }
    }
}
vlans {
    blue {
        vlan-id 100;
        interface xe-0/0/4.0:
        interface xe-0/0/6.0;
        l3-interface irb 100;
```

```
    }
    red {
      vlan-id 200;
      interface xe-0/0/0.0:
      interface xe-0/0/2.0;
      l3-interface irb 200;
    }
}
```

TIP: To quickly configure the blue and red VLAN interfaces, issue the load merge terminal command, copy the hierarchy, and paste it into the switch terminal window.

Verification

To verify that the **blue** and **red**VLANs have been created and are operating properly, perform these tasks:

- Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces on page 216
- Verifying That Traffic Can Be Routed Between the Two VLANs on page 216

Verifying That the VLANs Have Been Created and Associated with the Correct Interfaces

Purpose Verify that the VLANs **blue** and **red** have been created on the switch and that all connected interfaces on the switch are members of the correct VLAN.

Action List all VLANs configured on the switch:

```
user@switch> show vlans
Name          Tag      Interfaces
default                xe-0/0/0.0, xe-0/0/2.0, xe-0/0/4.0, xe-0/0/6.0,
blue          100       xe-0/0/4.0, xe-0/0/6,
red           200       xe-0/0/0.0, xe-0/0/2.0, *
mgmt                    me0.0*
```

Meaning The **show vlans** command lists all VLANs configured on the switch and which interfaces are members of each VLAN. This command output shows that the **blue** and **red** VLANs have been created. The **blue** VLAN has a tag ID of 100 and is associated with interfaces **xe-0/0/4.0** and **xe-0/0/6.0**. VLAN **red** has a tag ID of 200 and is associated with interfaces **xe-0/0/0.0** and **xe-0/0/2.0**.

Verifying That Traffic Can Be Routed Between the Two VLANs

Purpose Verify routing between the two VLANs.

Action Verify that the IRB logical units are up:

```
user@switch> show interfaces terse
irb.100                 up    up   inet     192.0.2.1/25
irb.200                 up    up   inet     192.0.2.129/25
```

NOTE: At least one port (access or trunk) with an appropriate VLAN assigned to it must be up for the irb interface to be up.

Verify that switch has created routes that use the IRB logical units:

```
user@switch> show route
192.0.2.0/25        *[Direct/0] 1d 03:26:45
                     > via irb.100
192.0.2.1/32        *[Local/0] 1d 03:26:45
                       Local via irb.100
192.0.2.128/25      *[Direct/0] 1d 03:26:45
                     > via irb.200
192.0.2.129/32      *[Local/0] 1d 03:26:45
                       Local via irb.200
```

List the Layer 3 routes in the switch's Address Resolution Protocol (ARP) table:

```
user@switch> show arp
MAC Address               Address           Name                      Flags

00:00:0c:06:2c:0d         192.0.2.7         irb.100                   None
00:13:e2:50:62:e0         192.0.2.132       irb.200                   None
```

Meaning The output of the **show interfaces** and **show route** commands show that the Layer 3 IRB logical units are working and the switch has used them to create direct routes that it will use to forward traffic between the VLAN subnets. The **show arp** command displays the mappings between the IP addresses and MAC addresses for devices on both **irb.100** (associated with VLAN **blue**) and **irb.200** (associated with VLAN **red**).These two devices can communicate.

Related Documentation

- Understanding Integrated Routing and Bridging on page 25
- *irb (Interfaces)*
- l3-interface on page 356

CHAPTER 15

VLAN Configuration Tasks

Configuring the Native VLAN Identifier

Switches support receiving and forwarding routed or bridged Ethernet frames with 802.1Q VLAN tags. The logical interface on which untagged packets are received must have the same native VLAN ID as that on the physical interface.

NOTE: This task uses Junos OS for QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring the Native VLAN Identifier (CLI Procedure)* .

To configure the native VLAN ID using the CLI:

1. Configure the port mode as **trunk** so that the interface is on multiple VLANs and can multiplex traffic between different VLANs. Trunk interfaces typically connect to other switches and to routers on the LAN.

   ```
   [edit interfaces xe-0/0/3 unit 0 family ethernet-switching]
   user@switch# set port-mode trunk
   ```

2. Configure the native VLAN ID:

   ```
   [edit interfaces xe-0/0/3 unit 0 family ethernet-switching]
   user@switch# set native-vlan-id 3
   ```

On a QFabric system, you can prevent packets with the native VLAN ID from being tagged by using the **except** configuration statement. Use this statement to specify that any egressing packet for the native VLAN of the configured interface will be untagged on egress.

This example shows how to configure a QFabric system to prevent tagging for native VLAN ID packets on egress:

```
set interfaces tor1:xe-0/0/0 unit 0 family ethernet-switching port-mode trunk
set interfaces tor1:xe-0/0/0 unit 0 family ethernet-switching vlan members all
```

```
set interfaces tor1:xe-0/0/0 unit 0 family ethernet-switching vlan except v1
set interfaces tor1:xe-0/0/0 unit 0 family ethernet-switching native-vlan-id v1
set vlans v1 vlan-id 1
set vlans v2 vlan-id 2
set vlans v3 vlan-id 3
```

This configuration defines VLAN **v1** as the native VLAN on interface **tor1:xe-0/0/0** and prevents egress traffic for that VLAN from being tagged. Without the **except v1** statement, packets would egress as tagged with VLAN ID 1.

Related Documentation

- Understanding Bridging and VLANs on page 5
- Example: Setting Up Bridging with Multiple VLANs on page 102
- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85

Configuring VLANs

Switches use VLANs to make logical groupings of network nodes with their own broadcast domains. You can use VLANs to limit the traffic flowing across the entire LAN and reduce collisions and packet retransmissions.

NOTE: This task uses Junos OS for the QFX Series that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring VLANs*.

For each endpoint on the VLAN, configure the following VLAN parameters on the corresponding interface:

1. Specify the description of the VLAN:

   ```
   [edit interfaces interface-name unit 0]
   user@switch# set description vlan-description
   ```

2. Specify the unique name of the VLAN:

   NOTE: In a QFabric system, do not configure “default” as the name of a VLAN. Though the QFabric system will allow you to configure and commit a VLAN with the name “default” in the current software with no commit errors, it will not work. Junos OS 12.2 and onwards will not allow you to commit a VLAN with the name “default.”

   ```
   [edit interfaces interface-name unit 0]
   user@switch# set family ethernet-switching vlan members vlan-name
   ```

3. Create the subnet for the VLAN:

   ```
   [edit interfaces]
   user@switch# set vlan unit 0 family inet address ip-address
   ```

4. Configure the VLAN tag ID or VLAN ID range for the VLAN:

   ```
   [edit vlans]
   user@switch# set vlan-name vlan-id vlan-id-number
   ```

or

```
[edit vlans]
user@switch# set vlan-name vlan-range vlan-id-low-vlan-id-high
```

5. Specify the maximum time that an entry can remain in the forwarding table before it ages out:

```
[edit vlans]
user@switch# set vlan-name mac-table-aging-time time
```

6. Specify a VLAN firewall filter to be applied to incoming or outgoing packets:

```
[edit vlans]
user@switch# set vlan-name filter (input | output) filter-name
```

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Configuring IRB Interfaces on page 257
- Creating a Series of Tagged VLANs on page 222
- Understanding Bridging and VLANs on page 5

Creating a Series of Tagged VLANs

When you divide an Ethernet LAN into multiple VLANs, each VLAN is assigned a unique IEEE 802.1Q tag. This tag is associated with each frame in the VLAN, and the network nodes receiving the traffic can use the tag to identify which VLAN a frame is associated with.

Instead of configuring VLANs and 802.1Q tags one at a time for a trunk interface, you can configure a VLAN range to create a series of tagged VLANs.

When an Ethernet LAN is divided into VLANs, each VLAN is identified by a unique 802.1Q tag. The tag is applied to all frames so that the network nodes receiving the frames can detect which VLAN the frames belong to. Trunk ports, which multiplex traffic among a number of VLANs, use the tag to determine the origin of frames and where to forward them.

For example, you could configure the VLAN **employee** and specify a tag range of **10 through 12**. This creates the following VLANs and tags:

- VLAN **employee-10**, tag **10**
- VLAN **employee-11**, tag **11**
- VLAN **employee-12**, tag **12**

Creating tagged VLANs in a series has the following limitations:

- Layer 3 interfaces do not support this feature.
- Because an access interface can only support one VLAN member, access interfaces also do not support this feature.

NOTE: This task uses Junos OS for QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does support ELS, see *Creating a Series of Tagged VLANs*..

To configure a series of tagged VLANs using the CLI (here, the VLAN is **employee**):

1. Configure the series (here, a VLAN series from 120 through 130):

   ```
   [edit]
   user@switch# set vlans employee vlan-range 120-130
   ```

2. Associate a series of tagged VLANs when you configure an interface in one of two ways:

- Include the name of the series:

  ```
  [edit interfaces]
  user@switch# set interfaces xe-0/0/22.0 family ethernet-switching vlanmembers employee
  ```

- Include the VLAN range:

  ```
  [edit interfaces]
  user@switch# set interfaces xe-0/0/22.0 family ethernet-switching vlan members 120–130
  ```

Associating a series of tagged VLANS to an interface by name or by VLAN range has the same result: VLANs **__employee_120__** through **__employee_130__** are created.

NOTE: When a series of VLANs is created using the vlan-range command, the VLAN names are preceded and followed by a double underscore.

Related Documentation

- Verifying That a Series of Tagged VLANs Has Been Created on page 369
- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Understanding Bridging*

CHAPTER 16

Unified Forwarding Table Configuration Task

Configuring the Unified Forwarding Table

To optimize the way your switch allocates memory for different types of addresses, you can choose a unified forwarding table profile. In addition to choosing this profile, you can also decide how you want memory allocated for longest prefix match (LPM) entries.

Configuring an Address-Storage Profile

On QFX5100 and EX4600 switches, you can control the allocation of memory available to store the following:

- MAC addresses
- Layer 3 host entries
- Longest prefix match (LPM) table entries

You configure the mix that best meets your needs by choosing the appropriate profile. Table 29 on page 225 lists the profiles you can choose and the maximum values for the MAC address and host table entries.

Table 29: Unified Forwarding Table Profiles

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K

Table 29: Unified Forwarding Table Profiles *(continued)*

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K
lpm-profile*	32K	16K	8K	8K	8K	4K	4K

Note that if the host table stores the maximum number of entries for any given type, the entire table is full and is unable to accommodate *any* entries of any other type. For example, an IPv6 unicast address occupies twice as much memory as an IPv4 unicast address, and an IPv6 multicast address occupies four times as much memory as an IPv4 unicast address. For more information about valid combinations of table entries see *Understanding the Unified Forwarding Table*.

To configure the profile that you want, enter and commit the following statement:

```
[edit]
user@switch# set chassis forwarding-options profile-name
```

NOTE: When you configure and commit a profile, the PFE process restarts and all the data interfaces on the switch go down and come back up.

The settings for **l2-profile-three** are configured by default. That is, if you do not enter a **set forwarding–options chassis *profile-name*** statement, these settings are configured.

Configuring the LPM Allocation

In addition to choosing a profile, you can further optimize memory allocation for LPM table entries by configuring how many IPv6 prefixes in the range /65 through /127 you want the switch to store. The switch uses LPM entries during address lookup to match addresses to the most-specific (longest) applicable prefix. The procedures for configuring the LPM table are different depending on which version of Junos OS you are using.

- Configuring the LPM Table With Junos OS 13.2X51-D10 and 13.2X52-D10 on page 226
- Configuring the LPM Table With Junos OS 13.2x51-D15 on page 228

Configuring the LPM Table With Junos OS 13.2X51-D10 and 13.2X52-D10

With Junos OS 13.2x51-D10 and 13.2X52-D10, the switch allocates memory for 16 IPv6 prefixes in the range /65 through /127 by default. If you want to use more than 16 IPv6 prefixes in this range, you must enter and commit the following statement:

```
user@switch# set chassis forwarding-options profile-name num-65-127-prefix [1-128]
```

Each increment adds support for 16 IPv6 prefixes between /65 and /127, for a maximum of 2048 such prefixes (16 x 128 = 2048). The system supports 16 of these prefixes by default, so to increase the number of supported prefixes, you must enter a value of 2 or

greater. For example, if you enter **2**, the system will support 32 IPv6 prefixes in the range /65 through /127.

NOTE: When you configure and commit the num-65-127-prefix value, all the data interfaces on the switch restart. The management interfaces are unaffected.

The LPM table is shared, and each increment that you add for IPv6 prefixes in the range /65 through /127 reduces the number of table entries that are available for IPv4 prefixes and IPv6 prefixes shorter than /65. Note that IPv6 prefixes /65 and longer consume twice as much memory as shorter IPv6 prefixes and four times as much memory as IPv4 prefixes. So, for example, entering the following statement

```
user@switch# set chassis forwarding-options l2-profile-one num-65-127-prefix 2
```

provides for 16 additional IPv6 prefixes /65 or longer (for a total of 32 such prefixes) and reduces the numbers of other prefixes that can be stored, as indicated:

- 32 fewer IPv6 prefixes shorter than /65 (16 IPv6 prefixes /65 or longer consume the same amount of memory as 32 IPv6 prefixes shorter than /65), or
- 64 fewer IPv4 prefixes (16 IPv6 prefixes /65 or longer consume the same amount of memory as 64 IPv4 prefixes)

Table 30 on page 227 provides examples of valid combinations that the LPM table can store using the **l2** and **l3** profiles. Once again, each row in the table represents a case in which the table is full and cannot accommodate any more entries.

Table 30: Example LPM Table Combinations Using l2-and l3 Profiles With Junos OS 13.2X51-D10 and 13.2X52-D10

num-65-127-prefix Value	IPv4 Entries	IPv6 Entries (Prefix <= 64)	IPv6 Entries (Prefix >= 65)
1 (default)	16K-16	0K	16
1 (default)	0K	8K-16	16
1 (default)	8K-16	4K	16
64	4K	4K	1K
64	2K	5K	1K
64	0K	6K	1K
128	4K	2K	2K
128	2K	3K	2K
128	0K	4K	2K

NOTE: With Junos OS 13.2X51-D10 and 13.2X52-D10, the **lpm-profile** does not support IPv6 prefixes. If you use this version of Junos OS and also use the **lpm-profile**, do not configure the **num-65-127-prefix** statement. That is, leave it at its default value of **1**, which allows for as many as 128K IPv4 prefixes (the maximum possible).

Configuring the LPM Table With Junos OS 13.2x51-D15

With Junos OS 13.2X51-D15, you can configure the memory allocation for the LPM table for the **lpm-profile** profile independently of the other profiles. In addition, Junos OS 13.2x51-D15 offers twice as much storage for IPv6 prefixes /65 through /127 (4K instead of 2K) for the **l2** and **l3** profiles.

- Configuring the l2 and l3 profiles With Junos OS 13.2x51-D15 on page 228
- Configuring The lpm-profile With Junos OS 13.2x51-D15 on page 229
- Configuring the lpm-profile With Junos OS 13.2x51-D40 and Later on page 229

Configuring the l2 and l3 profiles With Junos OS 13.2x51-D15

With Junos OS 13.2x51-D15, you can configure the switch to support as many as 4K IPv6 prefixes /65 through /127 if you are using any profile other than the **lpm-profile** profile. To do so, enter and commit the following statement:

```
user@switch# set chassis forwarding-options profile-name num-65-127-prefix [0-4]
```

Each increment adds support for 1K IPv6 prefixes between /65 and /127, for a maximum of 4K such prefixes. The default value is **1**, which allocates memory for 1K of IPv6 prefixes in this range. Each increment that you add for IPv6 prefixes in the range /65 through /127 reduces the number of table entries that are available for IPv6 prefixes shorter than /65 and IPv4 prefixes. Table 31 on page 228 shows the numbers of entries that you can allocate by using the **num-65-127-prefix** statement with Junos OS 13.2X51-D15. Once again, each row represents a case in which the table is full and cannot accommodate any more entries.

Table 31: LPM Table Combinations for l2 andl3 profiles With Junos OS 13.2X51-D15

num-65-127-prefix Value	IPv4 Entries	IPv6 Entries (Prefix <= 64)	IPv6 Entries (Prefix >= 65)
0	16K	8K	0K
1 (default)	12K	6K	1K
2	8K	4K	2K
3	4K	2K	3K
4	0K	0K	4K

NOTE: When you configure the **num-65-127-prefix** value, the PFE process restarts and all the data interfaces on the switch go down and come back up. The management interfaces are unaffected.

Configuring The lpm-profile With Junos OS 13.2x51-D15

If you use the **lpm-profile** profile with Junos OS 13.2x51-D15, you can control whether the switch allocates any memory for IPv6 prefixes /65 through /127. By default, the switch supports the following with this profile:

- 128K IPv4 prefixes
- 16K IPv6 prefixes (all lengths)

You can disable support for IPv6 prefixes /65 through /127 with the **lpm-profile** profile so that there is more memory for IPv6 prefixes shorter than /65. To do so, enter and commit the following statement:

```
user@switch# set chassis forwarding-options profile-name prefix-65-127-disable
```

If you enter this statement, the switch allocates memory for the following:

- 128K IPv4 and IPv6 prefixes shorter than /65
- 0K IPv6 prefixes /65 through /127

For example, if you use the **prefix-65-127-disable** statement, each of the following combinations are valid:

- 100K IPv4 and 28K IPv6 /64 prefixes
- 64K IPv4 and 64K IPv6 /64 prefixes
- 128K IPv4 and 0K IPv6 /64 prefixes
- 0K IPv4 and 128K IPv6 /64 prefixes

Configuring the lpm-profile With Junos OS 13.2x51-D40 and Later

If you use the **lpm-profile** profile with Junos OS 13.2x51-D40 or later, you can configure the system to store unicast IPv4 and IPv6 host addresses in the LPM table by using the **unicast-in-lpm** option, thereby freeing memory in the host table. When you use this option, unicast IPv4 and IPv6 addresses are stored in the LPM table instead of the host table, as shown in Table 32 on page 229. You can also use the **prefix-65-127-disable** option to maximize the number of IPv4 addresses and IPv6 addresses with prefixes shorter than /65 (and provide no memory for IPv6 addresses with prefixes longer than /64.)

Table 32: lpm-profile with unicast-in-lpm Option

prefix-65-127-disable?	MAC Table	Host Table (multicast addresses)						LPM Table unicast addresses)		
	MAC	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)	IPv4 unicast	IPv6 unicast (</65)	IPv6 unicast (>/64)

Table 32: lpm-profile with unicast-in-lpm Option *(continued)*

prefix-65-127-disable?	MAC Table	Host Table (multicast addresses)						LPM Table unicast addresses)		
No	32K	0	0	8K	8K	4K	4K	128K	16K	16K
Yes	32K	0	0	8K	8K	4K	4K	128K	128K	0

Note that all entries in each table share the same memory space. If a table stores the maximum number of entries for any given type, the entire shared table is full and is unable to accommodate any entries of any other type. For example, if you use the the **unicast-in-lpm** option and there are 128K IPv4 unicast addresses stored in the LPM table, the entire LPM table is full and no IPv6 addresses can be stored. Similarly, if you use the **unicast-in-lpm** option but do not use the **prefix-65-127-disable** option and 16K IPv6 addresses with prefixes shorter than /65 are saved, the entire LPM table is full and no additional addresses (IPv4 or IPv6) can be stored.

To use the **unicast-in-lpm** option, enter and commit the following statement:

```
user@switch# set chassis forwarding-options lpm-profile unicast-in-lpm
```

To use the **prefix-65-127-disable** option, enter and commit the following statement:

```
user@switch# set chassis forwarding-options lpm-profile prefix-65-127-disable
```

Related Documentation

- Understanding the Unified Forwarding Table on page 59

CHAPTER 17

Forwarding Mode Configuration Task

Configuring the Forwarding Mode

By default, packets packets are forwarded using store-and-forward mode. You can configure all the interfaces to use cut-through mode instead.

To enable cut-through switching mode, enter the following statement:

```
[edit forwarding-options]
user@switch# set cut-through
```

Related Documentation

- cut-through on page 281

CHAPTER 18

Interface Address Configuration Task

Configuring the Interface Address

You assign an address to an interface by specifying the address when configuring the protocol family. For the **inet** or **inet6** family, configure the interface IP address. For the **iso** family, configure one or more addresses for the loopback interface. For the **ccc**, **ethernet-switching**, **tcc**, **mpls**, **tnp**, and **vpls** families, you never configure an address.

NOTE: The point-to-point (PPP) address is taken from the loopback interface address that has the primary attribute. When the loopback interface is configured as an unnumbered interface, it takes the primary address from the donor interface.

To assign an address to an interface, include the **address** statement:

```
address address {
    broadcast address;
    destination address;
    destination-profile name;
    eui-64;
    preferred;
    primary;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number* **family** *family***]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number* **family** *family***]**

In the **address** statement, specify the network address of the interface.

For each address, you can optionally configure one or more of the following:

- Broadcast address for the interface subnet—Specify this in the **broadcast** statement; this applies only to Ethernet interfaces, such as the management interface **fxp0**, **em0**, or **me0** the Fast Ethernet interface, and the Gigabit Ethernet interface.
- Address of the remote side of the connection (for point-to-point interfaces only)—Specify this in the **destination** statement.
- PPP properties to the remote end—Specify this in the **destination-profile** statement. You define the profile at the **[edit access group-profile *name* ppp]** hierarchy level (for point-to-point interfaces only).
- Whether the router or switch automatically generates the host number portion of interface addresses—The **eui-64** statement applies only to interfaces that carry IPv6 traffic, in which the prefix length of the address is 64 bits or less, and the low-order 64 bits of the address are zero. This option does not apply to the loopback interface (**lo0**) because IPv6 addresses configured on the loopback interface must have a 128-bit prefix length.
- Whether this address is the preferred address—Each subnet on an interface has a preferred local address. If you configure more than one address on the same subnet, the preferred local address is chosen by default as the source address when you originate packets to destinations on the subnet.

  By default, the preferred address is the lowest-numbered address on the subnet. To override the default and explicitly configure the preferred address, include the **preferred** statement when configuring the address.

- Whether this address is the primary address—Each interface has a primary local address. If an interface has more than one address, the primary local address is used by default as the source address when you send packets from an interface where the destination provides no information about the subnet (for example, some **ping** commands).

By default, the primary address on an interface is the lowest-numbered non-127 (in other words, non-loopback) preferred address on the interface. To override the default and explicitly configure the preferred address, include the **primary** statement when configuring the address.

- Configuring Interface IPv4 Addresses on page 234
- Configuring Interface IPv6 Addresses on page 236

Configuring Interface IPv4 Addresses

You can configure router or switch interfaces with a 32-bit IP version 4 (IPv4) address and optionally with a destination prefix, sometimes called a *subnet mask*. An IPv4 address utilizes a 4-octet dotted decimal address syntax (for example, **192.16.1.1**). An IPv4 address with destination prefix utilizes a 4-octet dotted decimal address syntax with a destination prefix appended (for example, **192.16.1.1/30**).

To configure an IPv4 address on routers and switches running Junos OS, use the **edit interface *interface-name* unit *number* family inet address *a.b.c.d/nn*** statement at the **[edit interfaces]** hierarchy level.

NOTE: Juniper Networks routers and switches support /31 destination prefixes when used in point-to-point Ethernet configurations; however, they are not supported by many other devices, such as hosts, hubs, routers, or switches. You must determine if the peer system also supports /31 destination prefixes before configuration.

Operational Behavior of Interfaces when the Same IPv4 Address is Assigned to Them

You can configure the same IPv4 address on multiple physical interfaces. When you assign the same IPv4 address to multiple physical interfaces, the operational behavior of those interfaces differs, depending on whether they are implicitly or explicitly point-to-point .

NOTE: By default, all interfaces are assumed to be point-to-point (PPP) interfaces. For all interfaces except aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet, you can explicitly configure an interface to be a point-to-point connection.

The following examples show the sample configuration of assigning the same IPv4 address to implicitly and explicilty point-to-point interfaces, and their corresponding **show interfaces terse** command outputs to see their operational status.

Configuring same IPv4 address on implicitly PPP interfaces:

```
[edit]
user@host# show
ge-0/1/0 {
    unit 0 {
        family inet {
            address 200.1.1.1/24;
        }
    }
}

ge-3/0/1 {
    unit 0 {
        family inet {
            address 200.1.1.1/24;
        }
    }
}
```

The sample output shown below for the above configuration reveals that only **ge-0/1/0.0** was assigned the same IPv4 address **200.1.1.1/24** and its **link** state was **up**, while **ge-3/0/1.0** was not assigned the IPv4 address, though its **link** state was up, which means that it will be operational only when it gets a unique IPv4 address other than **200.1.1.1/24**.

```
user@host> show interfaces terse ge*
```

```
Interface               Admin Link Proto    Local                 Remote
        ge-0/1/0                up    up
        ge-0/1/0.0              up    up   inet     200.1.1.1/24
                                         multiservice
        ge-0/1/1                up    down
        ge-3/0/0                up    down
        ge-3/0/1                up    up
        ge-3/0/1.0              up    up   inet
                                         multiservice
```

Configuring same IPv4 address on explicitly PPP interfaces:

```
[edit]
user@host# show
so-0/0/0 {
  unit 0 {
    family inet {
      address 200.1.1.1/24;
    }
  }
}
so-0/0/3 {
  unit 0 {
    family inet {
      address 200.1.1.1/24;
    }
  }
}
```

The sample output shown below for the above configuration reveals that both **so-0/0/0.0** and **so-0/0/3.0** were assigned the same IPv4 address **200.1.1.1/24** and that their **link** states were down, which means that to make them operational atleast one of them will have be configured with a unique IPv4 address other than **200.1.1.1/24**.

```
user@host> show interfaces terse so*
Interface               Admin Link Proto    Local                 Remote
so-0/0/0                up    up
so-0/0/0.0              up    down inet     200.1.1.1/24
so-0/0/1                up    up
so-0/0/2                up    down
so-0/0/3                up    up
so-0/0/3.0              up    down inet     200.1.1.1/24
so-1/1/0                up    down
so-1/1/1                up    down
so-1/1/2                up    up
so-1/1/3                up    up
so-2/0/0                up    up
so-2/0/1                up    up
so-2/0/2                up    up
so-2/0/3                up    down
```

Configuring Interface IPv6 Addresses

NOTE: IPv6 is not currently supported for the QFX Series.

You represent IP version 6 (IPv6) addresses in hexadecimal notation using a colon-separated list of 16-bit values.

You assign a 128-bit IPv6 address to an interface by including the **address** statement:

address *aaaa*:*bbbb*:...:*zzzz*/*nn*;

NOTE: You cannot configure a subnet zero IPv6 address because RFC 2461 reserves the subnet-zero address for anycast addresses, and Junos OS complies with the RFC.

You can include this statement at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number* **family inet6]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number* **family inet6]**

The double colon (**::**) represents all bits set to 0, as shown in the following example:

```
interfaces fe-0/0/1 {
    unit 0 {
        family inet6 {
            address fec0:1:1:1::2/64;
        }
    }
}
```

NOTE: You must manually configure the router or switch advertisement and advertise the default prefix for autoconfiguration to work on a specific interface.

Related Documentation

- *Configuring IPCP Options*
- *Configuring Default, Primary, and Preferred Addresses and Interfaces*

CHAPTER 19

MAC Learning Configuration Tasks

Configuring MAC Notification

When a MAC address is learned or unlearned, SNMP notifications can be sent to the network management system at regular intervals to record the addition or removal of the MAC address. This process is known as *MAC notification*.

The MAC notification interval defines how often Simple Network Management Protocol (SNMP) notifications logging the addition or removal of MAC addresses on the switch are sent to the network management system.

MAC notification is disabled by default. When MAC notification is enabled, the default MAC notification interval is 30 seconds.

To enable or disable MAC notification, or to set the MAC notification interval, perform these tasks:

Enabling MAC Notification

MAC notification is disabled by default. You need to perform this procedure to enable MAC notification.

To enable MAC notification on the switch with the default MAC notification interval of 30 seconds:

```
[edit ethernet-switching-options]
user@switch# set mac-notification
```

To enable MAC notification on the switch with any other MAC notification interval (here, 60 seconds):

```
[edit ethernet-switching-options]
user@switch# set mac-notification notification-interval 60
```

Disabling MAC Notification

MAC notification is disabled by default. Perform this procedure only if MAC notification was previously enabled on your switch.

To disable MAC notification on the switch:

```
[edit ethernet-switching-options]
user@switch# delete mac-notification
```

Setting the MAC Notification Interval

The default MAC notification interval is 30 seconds. The procedure to change the MAC notification interval to a different interval is identical to the procedure to enable MAC notification on the switch with a nondefault value for the MAC notification interval.

To set the MAC notification interval on the switch (here, the MAC notification interval is set to 5 seconds):

```
[edit ethernet-switching-options]
user@switch# set mac-notification notification-interval 5
```

Related Documentation

- Verifying That MAC Notification Is Working Properly on page 369

Configuring MAC Table Aging

MAC table aging ensures that a switch tracks only active nodes on the network and that it is able to flush out network nodes that are no longer available.

To manage MAC entries more efficiently, you can configure an entry's aging time, which is the maximum time that an entry can remain in the MAC address table before it is deleted because it has reached its maximum age.

NOTE: This task uses Junos OS for Junos OS for QFX3500 and QFX3600 switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring MAC Table Aging*.

You can use the **set-mac-table-aging-time** command to configure how long entries remain in the Ethernet switching table before expiring. Here the VLAN is **employee-vlan**:

```
[edit vlans employee-vlan]
user@switch# set mac-table-aging-time 200
```

Related Documentation

- Understanding Bridging and VLANs on page 5
- Example: Setting Up Bridging with Multiple VLANs on page 102
- Example: Connecting an Access Switch to a Distribution Switch on page 76

Disabling MAC Learning

By default, MAC learning is globally enabled on all nodes in a device. This topic describes how to disable MAC learning, as well as how to reenable and verify that MAC learning has been enabled or disabled.

Disabling dynamic MAC learning on the device prevents a node from learning source and destination MAC addresses.

NOTE: This task uses Junos OS for QFX3500 and QFX3600 switches and does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Disabling MAC Learning*.

- To disable MAC learning on the QFX Series:

```
[edit ethernet-switching-options interfaces interface]
user@switch# set no-mac-learning
```

- To enable MAC learning on the QFX Series:

```
[edit ethernet-switching-options interfaces interface]
user@switch# delete no-mac-learning
user@switch# deactivate no-mac-learning
```

- To verify the status of MAC learning on the QFX Series, view the Ethernet MAC learning statistics in operational mode.

```
user@switch> show ethernet-switching table
Ethernet-switching table: 2 entries, 1 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  default           00:1f:12:39:90:80 Learn         29 xe-/0/0.0
```

Related Documentation

- Understanding MAC Learning on page 26
- Example: Disabling MAC Learning on page 109
- no-mac-learning on page 286

Disabling MAC Learning in a VLAN

By default, MAC learning is enabled on a VLAN. This topic describes how to disable MAC learning in a VLAN, as well as how to reenable and verify that MAC learning has been enabled or disabled.

Disabling dynamic MAC learning in a VLAN on a QFX Series product prevents a node from learning source and destination MAC addresses.

- To disable MAC learning in a VLAN:

```
[edit vlans vlan-name]
user@switch# set no-mac-learning
```

- To reenable MAC learning in a VLAN, use either of the following two commands:

```
[edit vlans vlan-name]
user@switch# delete no-mac-learning
user@switch# deactivate no-mac-learning
```

- To verify the status of MAC learning on the QFX series:

```
user@switch> show ethernet-switching table
```

Related Documentation

- Understanding MAC Learning on page 26
- Example: Disabling MAC Learning in a VLAN on page 110
- no-mac-learning (Per VLAN) on page 286

CHAPTER 20

Multiple VLAN Registration Protocol Configuration Task

Configuring Multiple VLAN Registration Protocol

Multiple VLAN Registration Protocol (MVRP) automates the creation and management of VLANs. When using MVRP on a QFabric system, you must manually create on the QFabric the VLANs that exist on the attached servers because the QFabric implementation of MVRP does not allow VLANs to be created dynamically. However, you do not need to manually assign VLAN membership to the QFabric ports that connect to the servers. MVRP automatically assigns VLAN membership to server-facing QFabric ports when it learns about a VLAN from an attached server. .

MVRP is disabled by default. To enable MVRP or set MVRP options, follow these instructions:

Enabling MVRP

MVRP can be enabled only on trunk interfaces. To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name
```

NOTE: On QFX Series switches, you must configure specific interfaces—you cannot specify interface all. You can enable MVRP on an interface range.

Disabling MVRP

MVRP is disabled by default. You only need to perform this procedure if you have previously enabled MVRP.

To disable MVRP on the entire QFabric system:

```
[edit protocols mvrp]
user@qfabric# set disable
```

To disable MVRP on a specific trunk interface:

```
[edit protocols mvrp]
user@qfabric# set disable interface interface-name
```

Configuring Timer Values

The timers in MVRP define the amount of time an interface waits to join or leave MVRP or to send or process the MVRP information for the switch after receiving an MVRP PDU. The join timer controls the amount of time the switch waits to accept a registration request, the leave timer controls the period of time that the switch waits in the Leave state before changing to the unregistered state, and the leaveall timer controls the frequency with which the LeaveAll messages are communicated.

The default MVRP timer values are 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms for the leaveall timer.

BEST PRACTICE: Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

To set the join timer for an interface:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name 300
```

To set the leave timer for an interface:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name leave-timer 1200
```

To set the leaveall timer for aninterface:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name leaveall-timer 12000
```

Configuring Passive Mode

QFX switches include a mode—called passive mode—in which an MVRP-configured interface does not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from a peer (server).

To configure an interface to operate in passive mode:

```
[edit protocols mvrp]
user@qfabric# set interface interface-name passive
```

Related Documentation

- Understanding Multiple VLAN Registration Protocol (MVRP) on page 47
- Example: Configuring Automatic VLAN Administration Using MVRP on page 71

- Verifying That MVRP Is Working Correctly on page 378

CHAPTER 21

Private VLAN Configuration Tasks

Creating a Private VLAN on a Single Switch

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature allows you to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a secondary VLAN inside a primary VLAN. This topic describes how to configure a PVLAN on a single switch.

Before you begin, configure names for all secondary VLANs that will be part of the primary VLAN. (You do not need to preconfigure the primary VLAN—it is configured as part of this procedure.) You do not need to create VLAN IDs (tags) for the secondary VLANs. It does not impair functioning if you tag the secondary VLANS, but tags are not used when secondary VLANs are configured on a single switch.

Keep these rules in mind when configuring a PVLAN:

- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN, you must first configure the primary VLAN and the PVLAN trunk port. You must also configure the primary VLAN to be private using the pvlan statement.
- If you are going to configure an isolated VLAN, you must first configure the primary VLAN and the PVLAN trunk port.

If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. To configure a private VLAN on a single switch:

1. Set the name and VLAN ID (802.1Q tag) for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id vlan-id-number
   ```
2. Configure the VLAN to be private:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name pvlan
   ```
3. Configure the trunk interfaces for the primary VLAN:

```
[edit interfaces]
user@switch# set interface-name unit 0 family ethernet-switching port-mode trunk
user@switch# set interface-name unit 0 family ethernet-switching vlan members
primary-vlan-name
```

4. Add the trunk interfaces to the primary VLAN:

```
[edit vlans]
user@switch# set primary-vlan-name interface interface-name
```

5. Configure the access interfaces for the community (secondary) VLANs:

```
[edit interfaces]
user@switch# set interface-name unit 0 family ethernet-switching port-mode access
```

6. Add the access interfaces to the community VLANs:

```
[edit vlans]
user@switch# set community-vlan-name interface interface-name
```

7. For each community VLAN, set the primary VLAN:

```
[edit vlans]
user@switch# set community-vlan-name primary-vlan primary-vlan-name
```

8. Configure isolated ports:

```
[edit vlans]
user@switch# set primary-vlan-name interface interface-name isolated
```

Related Documentation

- Understanding Private VLANs on page 28
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 33
- Creating a Private VLAN Spanning Multiple Switches on page 249
- Verifying That a Private VLAN Is Working on page 372

Creating a Private VLAN Spanning Multiple Switches

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic and to even limit the communication between known hosts. The private VLAN (PVLAN) feature allows you to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a secondary VLAN inside a primary VLAN. This topic describes how to configure a PVLAN to span multiple switches.

Before you begin, configure names for all secondary VLANs that will be part of the primary VLAN. (You do not need to preconfigure the primary VLAN—it is configured as part of this procedure.) You do not need to create VLAN IDs (tags) for the secondary VLANs. It does not impair functioning if you tag the secondary VLANS, but tags are not used when secondary VLANs are configured on a single switch.

The following rules apply to creating PVLANs:

- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN, you must first configure the primary VLAN and the PVLAN trunk port. You must also configure the primary VLAN to be private using the pvlan statement.
- If you are going to configure an isolated VLAN, you must first configure the primary VLAN and the PVLAN trunk port.

If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. To configure a private VLAN to span multiple switches:

1. Set the name and VLAN ID (802.1Q tag) for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id vlan-id-number
   ```
2. Configure the VLAN to be private:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name pvlan
   ```
3. Configure the trunk interfaces for the primary VLAN:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching port-mode trunk
   user@switch# set interface-name unit 0 family ethernet-switching vlan members
   primary-vlan-name
   ```
4. Add the trunk interfaces to the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name interface interface-name
   ```
5. Configure the access interfaces for the community (secondary) VLANs:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching port-mode access
   ```
6. Add the access interfaces to the community VLANs:

   ```
   [edit vlans]
   user@switch# set community-vlan-name interface interface-name
   ```
7. For each community VLAN, set the primary VLAN:

   ```
   [edit vlans]
   ```

```
user@switch# set community-vlan-name primary-vlan primary-vlan-name
```

8. Configure an isolated VLAN ID to create an interswitch isolated domain that spans the switches:

```
[edit vlans]
user@switch# set primary-vlan-name isolation-vlan-id number
```

9. Configure isolated ports:

```
[edit vlans]
user@switch# set primary-vlan-name interface interface-name isolated
```

Related Documentation

- Understanding Private VLANs on page 28
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 33
- Creating a Private VLAN on a Single Switch on page 247
- Verifying That a Private VLAN Is Working on page 372

CHAPTER 22

Proxy ARP Configuration Task

Configuring Proxy ARP

You can configure proxy Address Resolution Protocol (ARP) to enable the switch to respond to ARP queries for network addresses by offering its own media access control (MAC) address. With proxy ARP enabled, the switch captures and routes traffic to the intended destination.

To configure proxy ARP on a single interface:

```
[edit interfaces]
user@switch# set xe-0/0/3 unit 0 proxy-arp restricted
```

BEST PRACTICE: We recommend that you configure proxy ARP in restricted mode. In restricted mode, the switch is not a proxy if the source and target IP addresses are on the same subnet. If you use unrestricted mode, disable gratuitous ARP requests on the interface to avoid the situation of the switch's response to a gratuitous ARP request appearing to the host to be an indication of an IP conflict:

To configure proxy ARP on a routed VLAN interface (RVI):

```
[edit interfaces]
user@switch# set vlan unit 100 proxy-arp restricted
```

Related Documentation

- Understanding Proxy ARP on page 67
- Verifying That Proxy ARP Is Working Correctly on page 377
- Understanding Integrated Routing and Bridging on page 25

CHAPTER 23

Q-in-Q Tunneling Configuration Tasks

Configuring Q-in-Q Tunneling

Q-in-Q tunneling and VLAN translation allow service providers to create a Layer 2 Ethernet connection between two customer sites. Providers can segregate different customers' VLAN traffic on a link (for example, if the customers use overlapping VLAN IDs) or bundle different customer VLANs into a single service VLAN. Data centers can use Q-in-Q tunneling to isolate customer traffic within a single site or when customer traffic flows between cloud data centers in different geographic locations.

NOTE: This task uses a Junos OS release that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring Q-in-Q Tunneling*.

Before you begin setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs on the neighboring switches. See "Configuring VLANs" on page 220.

To configure Q-in-Q tunneling:

1. Create the service VLAN (S-VLAN) and configure an ID for it:

   ```
   [edit vlans]
   user@switch#  set s-vlan-name vlan-id s-vlan-ID
   ```

2. Enable Q-in-Q tunneling on the S-VLAN:

   ```
   [edit vlans]
   user@switch#  set s-vlan-name dot1q-tunneling
   ```

3. Set the allowed customer VLANs (C-VLANs) on the S-VLAN (optional). Here, the C-VLANs are identified by a range:

   ```
   [edit vlans]
   user@switch#  set s-vlan-name dot1q-tunneling customer-vlans range
   ```

4. Configure a global value for the tag protocol identifier (EtherType) of the service VLAN tags (optional):

   ```
   [edit]
   user@switch#  set ethernet-switching-options dot1q-tunneling ether-type ether-type-value
   ```

Depending on your interface configuration, you might need to adjust the MTU value on your trunk or access ports to accommodate the 4 bytes used for the tag added by Q-in-Q tunneling. For example, if you use the default MTU value of 1514 bytes on your access and trunk ports, you need to make one of the following adjustments:

- Reduce the MTU on the access links by at least 4 bytes so that the frames do not exceed the MTU of the trunk link when S-VLAN tags are added.
- Increase the MTU on the trunk link so that the link can handle the larger frame size.

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 61
- Example: Setting Up Q-in-Q Tunneling on page 151
- *Troubleshooting Q-in-Q and VLAN Translation Configuration*
- *mtu*

CHAPTER 24

Reflective Relay Configuration Task

Configuring Reflective Relay

Configure reflective relay when a switch port must return packets on a downstream port. For example, configure reflective relay when a switch port receives aggregated virtual machine packets from a technology such as virtual Ethernet port aggregator (VEPA). When these packets are passed through the switch, reflective relay allows the switch to send those packets back on the same interface that was used for delivery.

NOTE: This task uses Junos OS for QFX3500 and QFX3600 switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring Reflective Relay*.

Before you begin configuring reflective relay, ensure that you have:

- Configured packet aggregation on the server connected to the port. See your server documentation.
- Configured the port for all VLANs that could be included in aggregated packets..

To configure reflective relay:

1. Configure an Ethernet interface with a port mode of **tagged-acces**:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type port-mode
   tagged-access
   ```

   For example:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching port-mode
   tagged-access
   ```

2. Configure the interface for reflective relay:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type reflective-relay
   ```

   For example:

   ```
   [edit]
   ```

```
user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching reflective-relay
```

3. Configure the interface for the VLANs that exist on the VM server:

```
[edit]
user@switch# set interfaces interface-name unit number family family-type vlan members
vlan-names
```

For example:

```
[edit]
user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members
[VLAN_Purple VLAN_Orange VLAN_Blue]
```

Related Documentation

- Example: Configuring Reflective Relay for Use with VEPA Technology on page 155
- Understanding Reflective Relay for Use with VEPA Technology on page 27

CHAPTER 25

Routed VLAN Interface Configuration Task

Configuring IRB Interfaces

Integrated routing and bridging (IRB) interfaces enable a switch to recognize which packets are being sent to local addresses so that they are bridged whenever possible and are routed only when needed. Whenever packets can be switched instead of routed, several layers of processing are eliminated. Switching also reduces the number of address look-ups.

NOTE: In versions of Junos OS that do not support Enhanced Layer 2 Software (ELS), this type of interface is called a routed VLAN interface (RVI).

To configure the routed VLAN interface:

1. Create the VLAN by assigning it a name and a VLAN ID:

   ```
   [edit]
   user@switch# set vlans support vlan-id 111
   ```

2. Assign an interface to the VLAN by specifying the logical interface (with the **unit** statement) and specifying the VLAN name as the member:

   ```
   [edit]
   user@switch# set interfaces ge-0/0/18 unit 0 family ethernet-switching vlan members support
   ```

3. Create the subnet for the VLAN's broadcast domain:

   ```
   [edit]
   user@switch# set interfaces irb unit 111 family inet address 111.111.111.1/24
   ```

4. Bind a Layer 3 interface with the VLAN:

   ```
   [edit]
   user@switch# set vlans support l3-interface irb.111
   ```

NOTE: If you are using a version of Junos OS that does not support ELS, you create a Layer 3 virtual interface named vlan

NOTE: Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple VLANs. Within a VLAN, traffic is bridged, while across VLANs, traffic is routed.

You can display the configuration settings:

```
user@switch> show interfaces irb terse
Interface               Admin Link Proto    Local                 Remote
vlan                    up    up
irb.111                 up    up   inet     111.111.111.1/24

user@switch> show vlans
Name            Tag     Interfaces
default
                        None
employee-vlan   20
                        ge-1/0/0.0, ge-1/0/1.0, ge-1/0/2.0
marketing       40
                        ge-1/0/10.0, ge-1/0/20.0, ge-1/0/30.0
support         111
                        ge-0/0/18.0
mgmt
                        bme0.32769, bme0.32771*

user@switch> show ethernet-switching table
Ethernet-switching table: 1 entries, 0 learned
  VLAN              MAC address       Type         Age Interfaces
  support           00:19:e2:50:95:a0 Static         - Router
```

Related Documentation

- Understanding Integrated Routing and Bridging on page 25

CHAPTER 26

Spanning Tree Protocol Configuration Tasks

Configuring STP

The default spanning-tree protocol on the device is Rapid Spanning Tree Protocol (RSTP). RSTP provides faster convergence times than Spanning Tree Protocol (STP) does. However, some legacy networks require the slower convergence times of basic STP.

If your network includes 802.1D 1998 bridges, you can remove RSTP and explicitly configure STP. When you explicitly configure STP, the device uses the IEEE 802.1D 2004 specification, force version 0. This configuration runs a version of RSTP that is compatible with the classic, basic STP.

To configure STP using the CLI:

1. Delete the RSTP configuration on the interface (here, the interface is **xe-0/0/5**):

   ```
   [edit]
   user@switch# delete protocols rstp interface xe-0/0/5
   ```

2. Configure STP on the interface:

   ```
   [edit]
   user@switch# set protocols stp interface xe-0/0/5
   ```

3. Commit the configuration:

   ```
   [edit]
   user@switch# commit
   ```

Related Documentation

- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437
- Overview of Spanning-Tree Protocols on page 51

Unblocking an Interface That Receives BPDUs in Error

NOTE: BPDU block protection is disabled on Node devices.

Devices use bridge protocol data unit (BPDU) protection on interfaces to prevent them from receiving BPDUs that could trigger a spanning-tree misconfiguration. If BPDUs are received on a BPDU-protected interface, the interface transitions to a blocking state and stops forwarding frames.

After you fix the misconfiguration that triggered the sending of BPDUs to an interface, you can unblock the interface and return it to service.

NOTE: This task describes how to use both the original CLI and the Enhanced Layer 2 Software (ELS) CLI. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

To unblock an interface after fixing the misconfiguration that triggered the BPDUs and return it to service:

- (Original CLI) Automatically unblock an interface by configuring a timer that expires (here, the interface is **xe-0/0/6**):

```
[edit ethernet-switching-options]
user@switch# set bpdu-block disable-timeout 30 interface xe-0/0/6
```

- (ELS CLI) Automatically unblock an interface by configuring a timer that expires (here, the interface is **xe-0/0/6**):

```
[edit protocols layer2-control]
user@switch# set bpdu-block disable-timeout 30 interface xe-0/0/6
```

- Manually unblock an interface using the operational mode command:

```
user@switch> clear bpdu-error interface xe-0/0/6
```

- Verify that the interface has been unblocked using the operational command:

```
user@switch> show ethernet-switching interfaces interface xe-0/0/6
 Interface   State    VLAN members           Blocking
xe-0/0/6.0  down     default                unblocked
```

Related Documentation

- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55

Configuring VLAN Spanning Tree Protocol

VLAN Spanning Tree Protocol (VSTP) enables the device to run one or more Spanning Tree Protocol (STP) or Rapid Spanning Tree Protocol (RSTP) instances for each VLAN on which VSTP is enabled. For networks with multiple VLANs, VSTP improves intelligent tree spanning by defining the best paths within the VLANs instead of within the entire network.

NOTE: This task uses Junos OS software that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Configuring VLAN Spanning-Tree Protocol*.

To configure VSTP:

1. (Optional) Enable Rapid Spanning Tree Protocol (RSTP):

   ```
   [edit protocols]
   user@switch# set rstp
   ```

   VSTP can run on a maximum of 253 VLANs; RSTP runs on the remaining VLANs if configured. Enabling RSTP ensures that a spanning-tree protocol runs on all VLANs.

2. Enable VSTP.

   - To enable VSTP on multiple VLANs using a VLAN group:

     ```
     [edit protocols]
     user@switch# set vstp vlan-group group group-name vlan vlan-id-range
     ```

   - To enable VSTP on all VLANs:

     ```
     [edit protocols]
     user@switch# set vstp vlan all
     ```

     NOTE: You must enable RSTP if you used the set vstp vlan all statement to enable VSTP and if the switch has more than 253 VLANs. If you use the set vstp vlan all statement to enable VSTP on a switch with more than 253 VLANs, the configuration cannot be committed.

   - To enable VSTP on a VLAN using a single VLAN ID:

     ```
     [edit protocols]
     user@switch# set vstp vlan vlan-id
     ```

   - To enable VSTP on a VLAN using a single VLAN name:

     ```
     [edit protocols]
     user@switch# set vstp vlan vlan-name
     ```

Related Documentation

- Understanding VSTP on page 54

CHAPTER 27

Static ARP Entries Configuration Task

Configuring Static ARP Entries

You can create static ARP table entries, which are explicit mappings between IP addresses and MAC addresses.

- To configure a static ARP entry:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
user@switch# set arp ip-address (mac | multicast-mac) mac-address
```

The IP address that you specify must be part of the subnet defined in the enclosing **address** statement.

To associate a multicast MAC address with a unicast IP address, use the **multicast-mac** statement.

Specify the MAC address as 6 hexadecimal bytes in one of the following formats: *nnnn.nnnn.nnnn* or *nn:nn:nn:nn:nn:nn*; for example, **0011.2233.4455** or **00:11:22:33:44:55**.

Related Documentation

- *Understanding Static ARP Entries*
- arp on page 350

CHAPTER 28

Ethernet Switching Options Configuration Statements

ethernet-switching-options

Syntax

```
ethernet-switching-options {
  analyzer {
    name {
      input {
        egress {
          interface (all | interface-name);
        }
        ingress {
          interface (all | interface-name);
          vlan (vlan-id | vlan-name);
        }
      output {
        interface interface-name;
        ip-address ip-address;
        vlan (vlan-id | vlan-name);
      }
    }
  }
  bpdu-block {
    interface (all | [interface-name]);
    disable-timeout timeout;
  }
  dot1q-tunneling {
    ether-type (0x8100 | 0x88a8 | 0x9100)
  }
  interfaces interface-name {
    no-mac-learning;
  }
  mac-table-aging-time seconds {
  }
  port-error-disable {
    disable-timeout timeout;
  }
  secure-access-port {
    dhcp-snooping-file {
      location local_pathname | remote_URL;
      timeout seconds;
      write-interval seconds;
    }
    interface (all | interface-name) {
      allowed-mac {
        mac-address-list;
      }
      (dhcp-trusted | no-dhcp-trusted);
      fcoe-trusted;
      mac-limit limit action action;
      no-allowed-mac-log;
    }
    vlan (all | vlan-name) {
      (arp-inspection | no-arp-inspection) [
        forwarding-class (for DHCP Snooping or DAI Packets) class-name;
      }
```

```
        dhcp-option82 {
          circuit-id {
            prefix (Circuit ID for Option 82) hostname;
            use-interface-description;
            use-vlan-id;
          }
          remote-id {
            prefix (Remote ID for Option 82) hostname | mac | none;
            use-interface-description;
            use-string string;
          }
          vendor-id <string>;
        }
        (examine-dhcp | no-examine-dhcp) {
          forwarding-class (for DHCP Snooping or DAI Packets) class-name;
        }
        examine-fip {
          examine-vn2vn {
            beacon-period milliseconds;
          }
          fc-map fc-map-value;
          no-fip-snooping-scaling;
        }
        mac-move-limit limit <fabric-limit limit action action;
      }
    }
    static {
      vlan vlan-id {
        mac mac-address next-hop interface-name;
      }
    }
    storm-control {
      interface (all | interface-name) {
        bandwidth bandwidth;
        no-broadcast;
        no-multicast;
        no-unknown-unicast;
      }
    }
    traceoptions {
      file filename <files number> <no-stamp> <replace> <size size> <world-readable |
        no-world-readable>;
      flag flag <disable>;
    }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure Ethernet switching options.

The remaining statements are explained separately.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Understanding Port Mirroring*
- *Overview of Access Port Protection*
- *Understanding Storm Control*

interfaces

Syntax
```
interfaces interface-name {
  no-mac-learning;
}
```

Hierarchy Level
[edit ethernet-switching-options]

Release Information
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description
Configure settings for interfaces that have been assigned to family **ethernet-switching**.

Options
interface-name —Name of an interface that is configured for **family ethernet-switching**.

The remaining statement is explained separately.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

traceoptions (Ethernet Switching Options)

Syntax

```
traceoptions {
  file filename <files number> <no-stamp> <replace> <size size> <world-readable |
    no-world-readable>;
  flag flag <disable>;
}
```

Hierarchy Level [edit ethernet-switching-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

NOTE: The **traceoptions** statement is not supported on the QFX3000 QFabric system.

Description Define global tracing operations for access security features on Ethernet switches.

Default The **traceoptions** feature is disabled by default.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename* —Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

files *number* —(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached (***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify gigabytes), at which point the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

flag *flag* —Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **access-security**—Trace access security events.
- **all**—All tracing operations.
- **analyzer**—Trace analyzer events.
- **config-internal**—Trace internal configuration operations.
- **filter**—Trace filter transaction events.
- **forwarding-database**—Trace forwarding database events.

- **general**—Trace general events.
- **interface**—Trace interface events.
- **krt**—Trace communications over routing sockets.
- **lib**—Trace library calls.
- **nexthop**—Trace next-hop events.
- **normal**—Trace normal events.
- **parse**—Trace reading of the configuration.
- **regex-parse**—Trace regular-expression parsing operations.
- **rtg**—Trace redundant trunk group events.
- **state**—Trace state transitions.
- **stp**—Trace spanning-tree events.
- **task**—Trace Ethernet-switching task processing.
- **timer**—Trace Ethernet-switching timer processing.
- **unknown-unicast-forwarding**—Trace unknown unicast forwarding events.
- **vlan**—Trace VLAN events.

no-stamp—(Optional) Do not timestamp the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Restrict file access to the user who created the file.

replace—(Optional) Replace an existing trace file if there is one rather than appending to it.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size ***size*** —(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify gigabytes

Range: 10 KB through 1 gigabyte

Default: 128 KB

world-readable—(Optional) Enable unrestricted file access.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Overview of Spanning-Tree Protocols on page 51
- *Understanding Bridging*

unknown-unicast-forwarding

Syntax

```
unknown-unicast-forwarding {
  vlan (all | vlan-name){
    interface interface-name;
  }
}
```

Hierarchy Level

[edit ethernet-switching-options],
[edit switch-options]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure the switch to forward all unknown unicast packets in a VLAN or on all VLANs to a particular interface.

NOTE: Before you can configure unknown unicast forwarding within a VLAN, you must first configure that VLAN.

The remaining statements are explained separately.

Default

Unknown unicast packets are flooded to all interfaces that belong to the same VLAN.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Unknown Unicast Forwarding*
- *Understanding Unknown Unicast Forwarding*
- show ethernet-switching table on page 408
- show vlans on page 448

CHAPTER 29

Fabric Control Configuration Statements

fabric-control

Syntax

```
fabric-control {
  graceful-restart {
    restart-time seconds;
    stale-routes-time seconds;
  }
}
```

Hierarchy Level [edit fabric protocols]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Specify attributes for the fabric control protocol.

The remaining statements are explained separately.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- *Understanding Routing Engines in the QFabric System*

graceful-restart (Fabric Control)

Syntax

```
graceful-restart {
  restart-timeseconds;
  stale-routes-time seconds;
}
```

Hierarchy Level [edit fabric protocols fabric-control]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Configure graceful restart parameters for the fabric control in a QFabric system.

The remaining statements are explained separately.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- *Understanding Routing Engines in the QFabric System*

protocols (Fabric)

Syntax

```
protocols {
  fabric-control {
    graceful-restart {
      restart-timeseconds;
      stale-routes-time seconds;
    }
  }
}
```

Hierarchy Level [edit fabric]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Specify attributes for the fabric control protocol.

The remaining statements are explained separately.

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- *Understanding Routing Engines in the QFabric System*

restart-time (Fabric Control)

Syntax restart-time *seconds*;

Hierarchy Level [edit fabric protocols fabric-control graceful-restart]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Configure the duration of the graceful restart period for the fabric control Routing Engine.

The graceful restart resynchronization process takes longer when the QFabric system contains node groups that have a large number of VLANs. The graceful-restart duration should, therefore, be set higher when the QFabric system contains at least one node group with a large number of VLANs.

Configure a restart time of 600 seconds if the number of VLAN members (vmembers) exceeds 32k.

CAUTION: **Configuring the restart time restarts the session between the fabric control Routing Engine and the Node groups. Traffic is dropped as a result of this restart. Normal QFabric system operations should resume once the session has restarted without any further user actions.**

Options ***seconds***—Duration of the graceful restart period.

Default: 300 seconds

Range: 300 to 900 seconds

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- Understanding Bridging and VLANs on page 5
- *Understanding Routing Engines in the QFabric System*

stale-routes-time (Fabric Control)

Syntax stale-routes-time *seconds*;

Hierarchy Level [edit fabric protocols fabric-control graceful-restart]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Set the length of time that the fabric control Routing Engine waits to receive messages from devices before declaring them down. Configure a stale routes time of 1800 seconds if the number of VLAN members (vmembers) exceeds 32k.

Options ***seconds***—Amount of time that the fabric control Routing Engine waits to receive messages from other devices before declaring them down.

Default: 900 seconds

Range: 900 to 1800 seconds

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- Understanding Bridging and VLANs on page 5
- *Understanding Routing Engines in the QFabric System*

CHAPTER 30

Unified Forwarding Table Configuration Statements

forwarding-options (chassis)

Syntax

```
forwarding options profile-name {
    num-65-127-prefix value
    lpm-profile prefix-65-127-disable
}
```

Hierarchy Level [edit chassis]

Release Information Statement introduced in Junos 13.2 for the QFX Series.

Description Configure a unified forwarding table profile to allocate the amount a memory available for the following:

- MAC addresses
- Layer 3 host entries
- Longest prefix match table entries

Options **profile-name**—name of the profile to use for memory allocation in the unified forwarding table. Table 33 on page 278 lists the profiles you can choose and the associated values for each type of entry.

Table 33: Unified Forwarding Table Profiles

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
	MAC Addresses	IPv4 unicast	IPv6 unicast	IPv4 (*, G)	IPv4 (S, G)	IPv6 (*, G)	IPv6 (S, G)
l2-profile-one	288K	16K	8K	8K	8K	4K	4K
l2-profile-two	224K	80K	40K	40K	40K	20K	20K
l2-profile-three (default)	160K	144K	72K	72K	72K	36K	36K
l3-profile	96K	208K	104K	104K	104K	52K	52K
lpm-profile*	32K	16K	8K	8K	8K	4K	4K

* This profile supports only IPv4 in Junos OS 13.2X51-D10. With Junos OS 13.2X51-D15 it supports IPv4 and IPv6.

Note that if the host stores the maximum number of entries for any given type, the entire table is full and is unable to accommodate *any* entries of any other type. For information about valid combinations of table entries see "Understanding the Unified Forwarding Table" on page 59.

You configure the memory allocation for LPM table entries differently depending on whether you use Junos OS 13.2X51-D10 or Junos OS 13.2X51-D15 and later. To learn

how to configure memory allocation for LPM table entries see"Configuring the Unified Forwarding Table" on page 225.

Required Privilege Level

Related Documentation

- Understanding the Unified Forwarding Table on page 59
- Configuring the Unified Forwarding Table on page 225

num-65-127-prefix

Syntax num-65-127-prefix *value*

Hierarchy Level [edit chassis forwarding-options *profile-name*]

Release Information Statement introduced in Junos 13.2 for the QFX Series.

Description Configure the number of supported IPv6 prefixes in the range /65 through /127.

Options

value—With Junos OS 13.2X51D10: Value in the range 1 through 128. Each increment adds support for 16 IPv6 addresses with prefixes between /65 and /127, for a maximum of 2048 such addresses (16 x 128 = 2048).

value—With Junos OS 13.2X51D15: Value in the range 0 through 4. Each increment adds support for 1K IPv6 addresses with prefixes between /65 and /127, for a maximum of 4K such addresses.

Required Privilege Level

Related Documentation

- Configuring the Unified Forwarding Table on page 225

prefix-65-127-disable

Syntax prefix-65-127-disable

Hierarchy Level [edit chassis forwarding-options lpm-profile]

Release Information Statement introduced in Junos 13.2X51-D15 for the QFX Series.

Description Disable support in the longest prefix match (LPM) table for IPv6 prefixes in the range /65 through /127.

Required Privilege Level

Related Documentation

- Configuring the Unified Forwarding Table on page 225

CHAPTER 31

Forwarding Mode Configuration Statement

cut-through

Syntax cut-through;

Hierarchy Level [edit forwarding-options]

Description Configures all the interfaces in the QFX series switch or QFabric to use cut-through forwarding mode instead of store-and-forward mode.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Configuring the Forwarding Mode on page 231

CHAPTER 32

MAC Learning Configuration Statements

mac-limit

Syntax mac-limit *number*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the number of MAC addresses allowed on a VLAN.

Default MAC limit is disabled.

Options *number*—Maximum number of MAC addresses.
Range: 1 through 32768

NOTE: This statement is not supported on QFabric systems.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- show vlans on page 448
- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Configuring MAC Table Aging on page 240
- *Understanding Bridging*

mac-notification

Syntax

```
mac-notification {
  notification-interval seconds;
}
```

Hierarchy Level [edit ethernet-switching-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Enable MAC notification for a switch. If you configure this statement without setting a notification interval, MAC notification is enabled with the default MAC notification interval of 30 seconds.

The remaining statement is explained separately.

Default MAC notification is disabled by default.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring MAC Notification on page 239

mac-table-aging-time

Syntax mac-table-aging-time *seconds*;

Hierarchy Level For platforms without ELS:

[edit ethernet-switching-options],
[edit vlans *vlan-name*]

For platforms with ELS:

[edit vlans *vlan-name* switch-options]

Release Information Statement introduced for specific VLANs in Junos OS Release 11.1 for the QFX Series.

Description Define how long entries remain in the Ethernet switching table before expiring:

- If you specify this statement at the **[ethernet-switching-options]** hierarchy level, it applies to all VLANs on the switch.
- If you specify this statement at the **[vlans]** hierarchy level, it applies to the specified VLAN.

Default 300 seconds

Options *seconds*—Time that entries remain in the Ethernet switching table before being removed.

- **Range**—60 to 1,000,000 seconds.
- **Default**—300 seconds.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Configuring MAC Table Aging on page 240
- *Configuring MAC Table Aging*
- Understanding Bridging and VLANs on page 5
- **show ethernet-switching statistics aging on page 402**

no-mac-learning

Syntax no-mac-learning *limit*;

Hierarchy Level For platforms without ELS:

[edit ethernet-switching-options interfaces *interface-name*]

For platforms with ELS:

[edit vlans *vlan-name* switch-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Disable MAC address learning for the specified interface. Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member.

Default MAC learning is enabled.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

-

no-mac-learning (Per VLAN)

Syntax no-mac-learning *limit*;

Hierarchy Level [edit vlans *vlan-name*]
[edit vlans *vlan-name* switch-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Disables MAC address learning for the specified VLAN.

Default MAC learning is enabled.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

notification-interval

Syntax notification-interval *seconds*;

Hierarchy Level [edit ethernet-switching-options mac-notification]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the MAC notification interval for a switch.

The MAC notification interval is the amount of time the switch waits before sending learned or unlearned MAC address SNMP notifications to the network management server. For instance, if the MAC notification interval is set to 10, all of the MAC address addition and removal SNMP notifications are sent to the network management system every 10 seconds.

Options ***seconds***—The MAC notification interval, in seconds.

Range: 1 through 60

Default: 30

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring MAC Notification on page 239

CHAPTER 33

MVRP Configuration Statements

disable (MVRP)

Syntax disable;

Hierarchy Level [edit protocols mvrp],
[edit protocols mvrp interface(all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description Disable the MVRP configuration on the interface.

Default MVRP is disabled by default.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

interface (MVRP)

Syntax

```
interface (all | interface-name) {
  disable;
  join-timer milliseconds;
  leave-timer milliseconds;
  leaveall-timer milliseconds;
  registration (forbidden | normal);
}
```

Hierarchy Level [edit protocols mvrp]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description Specify interfaces on which to configure Multiple VLAN Registration Protocol (MVRP).

NOTE: On QFX Series switches, you must configure specific interfaces—you cannot specify **interface all**. You can enable MVRP on an interface range.

Default By default, MVRP is disabled.

Options **all**—All interfaces on the switch.

interface-name—Names of interface to be configured for MVRP.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

join-timer (MVRP)

Syntax join-timer *milliseconds*;

Hierarchy Level [edit protocols mvrp interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description Configure the maximum number of milliseconds interfaces must wait before sending Multiple VLAN Registration Protocol (MVRP) protocol data units (PDUs).

Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Default 200 milliseconds

Options ***milliseconds***—Number of milliseconds that the interface must wait before sending MVRP PDUs.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- leave-timer on page 292
- leaveall-timer on page 293
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

leave-timer (MVRP)

Syntax leave-timer *milliseconds*;

Hierarchy Level [edit protocols mvrp interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description For Multiple VLAN Registration Protocol (MVRP), configure the number of milliseconds the switch retains a VLAN in the Leave state before the VLAN is unregistered. If the interface receives a join message before this timer expires, the VLAN remains registered.

Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP.

Default 1000 milliseconds

Options ***milliseconds***—Number of milliseconds that the switch retains a VLAN in the Leave state before the VLAN is unregistered. At a minimum, set the leave-timer interval at twice the join-timer interval.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- join-timer on page 291
- leaveall-timer on page 293
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

leaveall-timer (MVRP)

Syntax leaveall-timer *interval*;

Hierarchy Level

- For platforms with ELS:

  [edit protocols mvrp],
  [edit protocols mvrp interface *interface-name*]

- For platforms without ELS:

  [edit protocols mvrp interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Hierarchy level **[edit protocols mvrp]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See *Getting Started with Enhanced Layer 2 Software* for information about ELS.)
Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description For Multiple VLAN Registration Protocol (MVRP), configure the interval at which the LeaveAll state operates on the interface.

Maintain default timer settings unless there is a compelling reason to change the settings. Modifying timers to inappropriate values might cause an imbalance in the operation of MVRP. However, if you choose to change the default values, keep in mind that on an EX Series switch that uses Junos OS with support for ELS, if the timer value set on an interface level is different from the value set on a switch level, then the value on the interface level takes precedence.

Options ***interval***—Number of seconds or milliseconds between the sending of Leave All messages.

Default: 10 seconds, or 10,000 milliseconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- join-timer on page 291
- leave-timer on page 292
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

passive (MVRP)

Syntax passive;

Hierarchy Level [edit protocols mvrp],
[edit protocols mvrp interface(all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 13.1 for the QFX Series.

Description Configure an MVRP interface to not announce its membership in a VLAN or send any VLAN declarations (updates) unless it receives registration for that VLAN from a peer (server).

Default Passive mode is disabled by default.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on page 71

CHAPTER 34

Private VLAN Configuration Statements

extend-secondary-vlan-id

Syntax extend-secondary-vlan-id *number*;

Hierarchy Level [edit vlans *vlan-name* pvlan]

Release Information Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description Configure traffic that egresses from a secondary VLAN trunk port to retain its secondary VLAN tag instead of getting the tag of the primary VLAN that the secondary port is a member of.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 37
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on page 139

isolated

Syntax isolated;

Hierarchy Level [edit vlans *vlan-name* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an access or trunk port to be isolated. You configure a trunk port to be isolated so that it can be a secondary VLAN trunk port–that is, it can carry secondary VLAN traffic.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249
- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 37
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on page 139

isolation-vlan-id

Syntax isolation-vlan-id *number*;

Hierarchy Level [edit vlans *vlan-name* pvlan]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an interswitch isolated VLAN within a private VLAN that spans multiple switches.

Options *number*—VLAN tag identifier.
Range: 0 through 4093

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

primary-vlan

Syntax primary-vlan *vlan-name*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure the primary VLAN for this community VLAN. The primary VLAN must be tagged, and the community VLAN must be untagged.

If you want to create a community VLAN, you must configure the primary VLAN to be private using the pvlan statement.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type ? after vlan or vlans in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

pvlan

Syntax pvlan;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify that the VLAN is private and access ports in the VLAN do not forward packets to each other. You use this statement with primary VLANs and isolated secondary VLANs.

Options none

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

promiscuous

Syntax promiscuous;

Hierarchy Level [edit vlans *vlan-name* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an access or trunk port to be promiscuous.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

pvlan-trunk

Syntax pvlan-trunk;

Hierarchy Level [edit vlans *vlan-name* interface]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an interface to be a private VLAN trunk port.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

vlans

Syntax

```
vlans {
  vlan-name {
    description text-description;
    dot1q-tunneling {
      customer-vlans (id | range);
    }
    filter input filter-name;
    filter output filter-name;
    interface interface-name {
      isolated;
      mapping (policy | tag push | native push);
      promiscuous;
    }
    isolation-vlan-id;
    l3-interface vlan.logical-interface-number;
    mac-limit number;
    mac-table-aging-time seconds;
    no-local-switching;
    no-mac-learning;
    primary-vlan vlan-name;
    pvlan extend-secondary-vlan-id vlan-id;
    vlan-id number;
    vlan-range vlan-id-low-vlan-id-high;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statements for private VLANs and Q-in-Q tunneling introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure VLAN properties on the QFX Series.

Default If you use the default factory configuration, all switch interfaces become part of the VLAN **default**.

Options *vlan-name*—Name of the VLAN. The name can contain letters, numbers, hyphens (-), and periods (.) and can be up to 255 characters long.

The remaining statements are described separately.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Configuring VLANs on page 220
- Configuring Q-in-Q Tunneling on page 253
- Creating a Series of Tagged VLANs on page 222

- Configuring IRB Interfaces on page 257
- Creating a Private VLAN on a Single Switch on page 247
- *Understanding Bridging*

CHAPTER 35

Proxy ARP Configuration Statement

proxy-arp

Syntax proxy-arp (restricted | unrestricted);

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number*],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.6 for EX Series switches.
restricted added in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description For Ethernet interfaces only, configure the router or switch to respond to any ARP request, as long as the router or switch has an active route to the ARP request's target address.

NOTE: You must configure the IP address and the inet family for the interface when you enable proxy ARP.

Default Proxy ARP is not enabled. The router or switch responds to an ARP request only if the destination IP address is its own.

Options

- **none**—The router or switch responds to any ARP request for a local or remote address if the router or switch has a route to the target IP address.
- **restricted**—(Optional) The router or switch responds to ARP requests in which the physical networks of the source and target are different and does not respond if the source and target IP addresses are in the same subnet. The router or switch must also have a route to the target IP address.
- **unrestricted**—(Optional) The router or switch responds to any ARP request for a local or remote address if the router or switch has a route to the target IP address.

Default: unrestricted

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Restricted and Unrestricted Proxy ARP*
- *Configuring Proxy ARP (CLI Procedure)*
- *Configuring Proxy ARP (CLI Procedure)*
- *Example: Configuring Proxy ARP on an EX Series Switch*
- *Configuring Gratuitous ARP*

CHAPTER 36

Q-in-Q Tunneling Configuration Statements

customer-vlans

Syntax customer-vlans (*id* | native | *range*);

Hierarchy Level [edit vlans *vlan-name* dot1q-tunneling]

Release Information Statement introduced in Junos OS Release 9.3 for EX Series switches.
Option **native** introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Limit the set of accepted customer VLAN tags to a range or to discrete values when mapping customer VLANs to service VLANs.

Options *id*—Numeric identifier for a VLAN.

native—Accepts untagged and priority-tagged packets from access interfaces and assigns the configured S-VLAN to the packet.

range—Range of numeric identifiers for VLANs. On the QFX series, you can include as many as eight separate customer VLAN ranges for a given service VLAN. Do not configure more than this number of ranges.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- **dot1q-tunneling (Ethernet Switching) on page 305**
- *ether-type*
- *Example: Setting Up Q-in-Q Tunneling on EX Series Switches*
- *Configuring Q-in-Q Tunneling (CLI Procedure)*
- *Understanding Q-in-Q Tunneling on EX Series Switches*
- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151
- **dot1q-tunneling (Ethernet Switching) on page 305**
- **ether-type on page 307**

dot1q-tunneling (Ethernet Switching)

Syntax

```
dot1q-tunneling {
  ether-type (0x8100 | 0x88a8 | 0x9100);
}
```

Hierarchy Level [edit ethernet-switching-options]

Release Information Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Set a global value for the EtherType for Q-in-Q tunneling.

The remaining statement is explained separately.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- **dot1q-tunneling on page 306**
- *Example: Setting Up Q-in-Q Tunneling on EX Series Switches*
- *Configuring Q-in-Q Tunneling (CLI Procedure)*
- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151
- **dot1q-tunneling on page 306**

dot1q-tunneling (VLANs)

Syntax

```
dot1q-tunneling {
  customer-vlans (id | native | range);
  layer2-protocol-tunneling all | protocol-name {
    drop-threshold number;
    shutdown-threshold number;
  }
}
```

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.3 for EX Series switches.
Option **native** introduced in Junos OS Release 9.6 for EX Series switches.
Options **layer2-protocol-tunneling**, **drop-threshold**, and **shutdown-threshold** introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Enable Q-in-Q tunneling on the specified VLAN.

NOTE:

- The VLAN on which you enable Q-in-Q tunneling must be a tagged VLAN.
- You cannot configure 802.1X user authentication on interfaces that have been enabled for Q-in-Q tunneling.

The remaining statements are explained separately.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Example: Setting Up Q-in-Q Tunneling on EX Series Switches*
- *Configuring Q-in-Q Tunneling (CLI Procedure)*
- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151
- *Configuring Layer 2 Protocol Tunneling*
- **dot1q-tunneling (Ethernet Switching) on page 305**

ether-type

Syntax ether-type (0x8100 | 0x88a8 | 0x9100)

Hierarchy Level [edit ethernet-switching-options dot1q-tunneling]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure a global value for the tag protocol identifier (EtherType) of the service VLAN tags (outer tags) in Q-in-Q tunneling. Only one EtherType value is supported at a time.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151

mapping

Syntax

mapping (native (push | swap) | *tag* (push | swap));
mapping native inner-tag *tag* push;
mapping native push inner-tag *tag*;

Hierarchy Level

[edit vlans *vlan-name* interface *interface-name* egress],
[edit vlans *vlan-name* interface *interface-name* ingress],
[edit vlans *vlan-name* interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Map a specific C-VLAN to an S-VLAN. By default, the received incoming or outgoing tag is replaced with the new tag.

Options

inner-tag (QFabric systems only)—apply the specified tag as an inner tag to packets that are received as untagged on an access interface.

native—Map untagged and priority-tagged packets to an S-VLAN.

push—Retain the incoming tag (as an inner tag) and adds an additional VLAN tag. When you use this option, the TPID of the outer tag is set as follows:

- If Q-in-Q tunneling is not enabled in the VLAN, then the Ethertype for outer tag is set to 0x8100.
- If Q-in-Q tunneling is enabled in the VLAN and a packet is egressing from a trunk port, then the Ethertype is set to 0x88a8 (or as configured by an **ether-type**statement).

swap—Replaces the incoming VLAN tag with the VLAN ID tag of the S-VLAN. Using this option is also referred to as VLAN ID translation. When you use this option on a trunk port for which Q-in-Q tunneling is enabled, use the ether-type statement to set the Ethertype.

tag—Original VLAN tag that will be replaced (with **swap**) or that will become an inner tag (with **push**).

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151

mapping-range

Syntax mapping-range *C-VLAN-range* (push | swap) <vlan-id-start *S-VLAN-ID*>;

Hierarchy Level [edit vlans *vlan-name* interface (VLANs) *interface-name*]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an access interface to map a range of C-VLANs to a range of S-VLANs. Use this statement instead of using multiple **set vlans *VLAN-name* interface *interface-name* mapping (push | swap)** statements to configure Q-in-Q tunneling or VLAN translation on a per-VLAN basis. This statement is particularly useful if you have used the **vlan-range** statement to create multiple VLANs.

Options

push—Retain the incoming tag and adds an additional VLAN tag (Q-in-Q tunneling).

swap—Swap the incoming VLAN tag with the VLAN ID tag of the S-VLAN (VLAN translation).

vlan-ID-start *S-VLAN-ID*—(Optional) Set the start of the S-VLAN range that the C-VLANs will be mapped to. If you omit this option, mapping begins with the first ID in the range of S-VLAN IDs (which you configure using the **set vlans** vlan-range statement).

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151
- **vlan-range on page 363**

no-local-switching

Syntax no-local-switching;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify that access ports in this VLAN domain do not forward packets to one another. You use this statement with primary VLANs and isolated secondary VLANs.

Required Privilege Level
routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

vlan-id-start

Syntax vlan-id-start *S-VLAN-ID*;

Hierarchy Level [edit vlans *vlan-name* interface (VLANs) *interface-name* mapping-range *C-VLAN-range* (push | swap)]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an access interface to map a range of C-VLANs to a range of S-VLANs. Use this statement instead of using multiple **set vlans *VLAN-name* interface *interface-name* mapping (push | swap)** statements to configure Q-in-Q tunneling or VLAN translation on a per-VLAN basis. This statement sets the start of the S-VLAN range that the C-VLANs are mapped to. If you omit this option, mapping begins with the first ID in the range of S-VLAN IDs (which you configure using the **set vlans** vlan-rangestatement).

Options None

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Configuring Q-in-Q Tunneling on page 253
- Example: Setting Up Q-in-Q Tunneling on page 151

vlans

Syntax

```
vlans {
  vlan-name {
    description text-description;
    dot1q-tunneling {
      customer-vlans (id | range);
    }
    filter input filter-name;
    filter output filter-name;
    interface interface-name {
      isolated;
      mapping (policy | tag push | native push);
      promiscuous;
    }
    isolation-vlan-id;
    l3-interface vlan.logical-interface-number;
    mac-limit number;
    mac-table-aging-time seconds;
    no-local-switching;
    no-mac-learning;
    primary-vlan vlan-name;
    pvlan extend-secondary-vlan-id vlan-id;
    vlan-id number;
    vlan-range vlan-id-low-vlan-id-high;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statements for private VLANs and Q-in-Q tunneling introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure VLAN properties on the QFX Series.

Default If you use the default factory configuration, all switch interfaces become part of the VLAN **default**.

Options ***vlan-name***—Name of the VLAN. The name can contain letters, numbers, hyphens (-), and periods (.) and can be up to 255 characters long.

The remaining statements are described separately.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Configuring VLANs on page 220
- Configuring Q-in-Q Tunneling on page 253
- Creating a Series of Tagged VLANs on page 222

- Configuring IRB Interfaces on page 257
- Creating a Private VLAN on a Single Switch on page 247
- *Understanding Bridging*

CHAPTER 37

Reflective Relay Configuration Statement

reflective-relay

Syntax reflective-relay;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure a switch interface to return packets back to a device on the same interface that was used to deliver the packets.

Default Switch interfaces are not configured for reflective relay.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Reflective Relay for Use with VEPA Technology on page 155
- Configuring Reflective Relay on page 255

CHAPTER 38

Spanning Tree Protocol Configuration Statements

- stp on page 342
- traceoptions (STP) on page 343
- vlan (STP) on page 347
- vstp on page 348

alarm (STP)

Syntax alarm;

Hierarchy Level [edit protocols mstp interface (all | *interface-name*) bpdu-timeout-action],
[edit protocols rstp interface (all | *interface-name*) bpdu-timeout-action],
[edit protocols stp interface (all | *interface-name*) bpdu-timeout-action],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*) bpdu-timeout-action]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For interfaces configured for loop protection, configure the software to generate a message to be sent to the system log file to record the loop-protection event.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Understanding Loop Protection for STP, RSTP, VSTP, and MSTP on page 56
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

block

Syntax block;

Hierarchy Level [edit protocols mstp (Spanning Trees) interface (all | *interface-name*) bpdu-timeout-action],
[edit protocols rstp interface (all | *interface-name*) bpdu-timeout-action],
[edit protocols stp interface (all | *interface-name*) bpdu-timeout-action],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*) bpdu-timeout-action]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure loop protection on a specific interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Understanding Loop Protection for STP, RSTP, VSTP, and MSTP on page 56
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

bpdu-block

Syntax

```
bpdu-block {
  interface (all | [interface-name]);
  disable-timeout timeout;
}
```

Hierarchy Level

- For platforms with ELS CLI:

  [edit protocols layer2-control]

- For platforms with Original CLI:

  [edit ethernet-switching-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure BPDU protection on an interface. If the interface receives BPDUs, it is disabled.

NOTE: **BPDU block protection is disabled on Node devices.**

The statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Unblocking an Interface That Receives BPDUs in Error on page 260
- **clear bpdu-error on page 382**
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

bpdu-block-on-edge

Syntax bpdu-block-on-edge;

Hierarchy Level [edit protocols mstp],
[edit protocols rstp],
[edit protocols vstp]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure bridge protocol data unit (BPDU) protection on all edge ports of a switch.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding VSTP on page 54
- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- **clear bpdu-error on page 382**
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

bpdu-timeout-action

Syntax

```
bpdu-timeout-action {
  alarm;
  block;
}
```

Hierarchy Level

[edit protocols mstp interface (all | *interface-name*)],
[edit protocols rstp interface (all | *interface-name*)],
[edit protocols stp interface (all | *interface-name*)],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure the BPDU timeout action on a specific interface. You must configure at least one action (**alarm**, **block**, or both).

The remaining statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Loop Protection to Prevent Interfaces from Transitioning from Blocking to Forwarding in a Spanning Tree on page 180
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding Loop Protection for STP, RSTP, VSTP, and MSTP on page 56
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

bridge-priority

Syntax bridge-priority *priority*;

Hierarchy Level [edit protocols mstp],
[edit protocols mstp *msti-id*],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the bridge priority. The bridge priority determines which bridge is elected as the root bridge. If two bridges have the same path cost to the root bridge, the bridge priority determines which bridge becomes the designated bridge for a LAN segment.

Options *priority*—Bridge priority. It can be set only in increments of 4096.

Range: 0 through 61,440

Default: 32,768

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

cost (STP)

Syntax cost *cost*;

Hierarchy Level [edit protocols mstp interface (all | *interface-name*)],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols rstp (Spanning Trees) interface (all | *interface-name*)],
[edit protocols stp interface (all | *interface-name*)],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), configure the link cost to control which bridge is the designated bridge and which interface is the designated interface.

Default Link cost is determined by the link speed.

Options ***cost***—Link cost associated with the port.

Range: 1 through 200,000,000

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

configuration-name (MSTP)

Syntax configuration-name *configuration-name*;

Hierarchy Level [edit protocols mstp]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Specify the configuration name. The configuration name is the MSTP region name carried in the MSTP BPDUs.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

disable (STP)

Syntax disable;

Hierarchy Level
[edit protocols mstp],
[edit protocols mstpinterface *interface-name*],
[edit protocols mstp msti *msti-id* vlan (*vlan-id* | *vlan-name*) interface *interface-name*],
[edit protocols rstp],
[edit protocols rstp interface *interface-name*],
[edit protocols stp],
[edit protocols stp interface *interface-name*],
[edit protocols vstp],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Disable STP, MSTP, RSTP, or VSTP on the switch or on a specific interface.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

disable-timeout (BPDU)

Syntax disable-timeout *timeout*;

Hierarchy Level [edit ethernet-switching-options bpdu-block]
[edit protocols layer2-control bpdu-block]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For interfaces configured for BPDU protection, specify the amount of time an interface receiving BPDUs is disabled.

Default The disable timeout is not enabled.

Options ***timeout***: Length of time, in seconds, that the interface receiving BPDUs is disabled. Once the timeout expires, the interface is brought back into service.

Range: 10 through 3600 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

edge (STP)

Syntax edge;

Hierarchy Level [edit protocols mstp interface (all | *interface-name*)],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols rstp interface (all | *interface-name*)],
[edit protocols stp interface (all | *interface-name*)],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), configure interfaces as edge interfaces. Edge interfaces immediately transition to a forwarding state.

Default Edge interfaces are not enabled.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

force-version

Syntax force-version stp;

Hierarchy Level [edit protocols vstp]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Force VLAN Spanning Tree Protocol (VSTP) to use the STP protocol instead of the default protocol, RSTP.

Options **stp**—Spanning Tree Protocol

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- show spanning-tree bridge on page 432
- *show spanning-tree interface*
- Understanding VSTP on page 54

forward-delay

Syntax forward-delay *seconds*;

Hierarchy Level [edit protocols mstp],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), specify how long a bridge interface remains in the listening and learning states before transitioning to the forwarding state.

Options ***seconds***—Number of seconds the bridge interface remains in the listening and learning states.

Range: 4 through 30 seconds

Default: 15 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

hello-time

Syntax hello-time *seconds*;

Hierarchy Level [edit protocols mstp],
[edit protocols rstp],
[edit protocols rstp],
[edit protocols vstp vlan *vlan-id*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), specify the time interval at which the root bridge transmits configuration BPDUs.

Options ***seconds***—Number of seconds between transmissions of configuration BPDUs.

Range: 1 through 10 seconds

Default: 2 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

interface (Spanning Trees)

Syntax

```
interface interface-name {
  arp-on-stp;
  bpdu-timeout-action
    block;
    log;
  cost cost;
  disable;
  edge;
  mode mode;
  no-root-port;
  priority priority;
}
```

Hierarchy Level

[edit protocols mstp],
[edit protocols mstp msti *msti-id*],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan (all | *vlan-id* | *vlan-name*)]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement updated in Junos OS Release 9.4 for EX Series switches to add VSTP support.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), configure an interface.

The **edge**, **mode**, and **no-root-port** options are not available at the [**edit protocols mstp msti *msti-id***] hierarchy level.

Options

interface-name—Name of an interface.

The remaining statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *show spanning-tree bridge*
- *show spanning-tree interface*
- *Example: Configuring Network Regions for VLANs with MSTP on EX Series Switches*
- *Example: Faster Convergence and Improved Network Stability with RSTP on EX Series Switches*
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- *Configuring VSTP (CLI Procedure)*

- show spanning-tree bridge on page 432

interface (BPDU)

Syntax interface (all | *interface-name*);

Hierarchy Level [edit ethernet-switching-options bpdu-block]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Apply BPDU protection to all interfaces or one or more interfaces.

Options **all**—All interfaces.

interface-name—Name of the interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55
- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437

interface (STP)

Syntax

```
interface interface-name {
  disable;
  cost cost;
  edge;
  mode mode;
  no-root-port;
  priority priority;
}
```

Hierarchy Level

[edit protocols mstp],
[edit protocols mstp msti],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), configure an interface.

Options

interface-name—Name of a Gigabit Ethernet interface.

The remaining statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding RSTP on page 53
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- *show spanning-tree interface*

max-age

Syntax max-age *seconds*;

Hierarchy Level [edit protocols mstp],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), specify the maximum age of received protocol BPDUs.

Options *seconds*—Maximum age of received protocol BPDUs.

Range: 6 through 40 seconds

Default: 20 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

max-hops

Syntax max-hops *hops*;

Hierarchy Level [edit protocols mstp]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Multiple Spanning Tree Protocol (MSTP), configure the maximum number of hops that a BPDU can be forwarded in the MSTP region.

Options ***hops*** — Number of hops the BPDU can be forwarded.

Range: 1 through 255 hops

Default: 20 hops

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

mode (STP)

Syntax mode *mode*;

Hierarchy Level
[edit protocols mstp interface (all | *interface-name*)],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols rstp interface (all | *interface-name*)],
[edit protocols stp interface (all | *interface-name*)],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), configure the link mode to identify point-to-point links.

Default For a full-duplex link, the default link mode is **point-to-point**. For a half-duplex link, the default link mode is **shared**.

Options *mode*—Link mode:

- **point-to-point**—Link is point to point.
- **shared**—Link is shared media.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

msti

Syntax

```
msti msti-id {
  vlan (vlan-id | vlan-name);
  interface interface-name {
    disable;
    cost cost;
    edge;
    mode mode;
    priority priority;
  }
}
```

Hierarchy Level [edit protocols mstp]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the Multiple Spanning Tree Instance (MSTI) identifier for Multiple Spanning Tree Protocol (MSTP). MSTI IDs are local to each region, so you can reuse the same MSTI ID in different regions.

Default MSTI is disabled.

Options *msti-id* —MSTI identifer.

Range: 1 through 4094. The Common Instance Spanning Tree (CIST) is always MSTI 0.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52

mstp

Syntax

```
mstp {
  disable;
  bpdu-timeout-action;
  bridge-priority priority;
  configuration-name (MSTP) name;
  forward-delay seconds;
  hello-time seconds;
  interface (all | interface-name) {
    bpdu-timeout-action {
      block;
      alarm;
    }
    disable;
    cost cost;
    edge;
    mode mode;
    no-root-port;
    priority priority;
  }
  max-age seconds;
  max-hops hops;
  msti msti-id {
    vlan (vlan-id | vlan-name);
    interface interface-name {
      disable;
      cost cost;
      edge;
      mode mode;
      priority priority;
    }
  }
  traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <(world-readable | no-world-readable)>;
    flag flag <flag-modifier> <disable>;
  }
  revision-level revision-level;
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure Multiple Spanning Tree Protocol (MSTP). MSTP is defined in the IEEE 802.1Q-2003 specification and is used to create a loop-free topology in networks with multiple spanning-tree regions.

The statements are explained separately.

Default MSTP is disabled.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52
- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437

priority (STP)

Syntax
priority *priority*;

Hierarchy Level
[edit protocols mstp interface (all | *interface-name*)],
[edit protocols mstp msti *msti-id* interface *interface-name*],
[edit protocols rstp interface (all | *interface-name*)],
[edit protocols stp interface (all | *interface-name*)],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description
For Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), VLAN Spanning Tree Protocol (VSTP), or Multiple Spanning Tree Protocol (MSTP), specify the interface priority to control which interface is elected as the root port.

Options
priority—Interface priority. The interface priority must be set in increments of 16.
Range: 0 through 240
Default: 128

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437

no-root-port

Syntax no-root-port;

Hierarchy Level
[edit protocols mstp interface (all | *interface-name*)],
[edit protocols rstp interface (all | *interface-name*)],
[edit protocols stp interface (all | *interface-name*)],
[edit protocols vstp vlan *vlan-id* interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure an interface to be a spanning tree designated port. If the bridge receives more STP bridge protocol data units (BPDUs) on a root-protected interface, that interface transitions to a root-prevented STP state (inconsistency state) and the interface is blocked. This blocking prevents a bridge that should not be the root bridge from being elected the root bridge. When the bridge stops receiving more STP BPDUs on the root-protected interface, interface traffic is no longer blocked.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Root Protection to Enforce Root Bridge Placement in Spanning Trees on page 207
- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

revision-level

Syntax revision-level *revision-level*;

Hierarchy Level [edit protocols mstp]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Multiple Spanning Tree Protocol (MSTP), set the revision number of the MSTP configuration.

Default The revision number is disabled.

Options ***revision-level***—Revision number of the MSTP region configuration.

Range: 0 through 65535

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52
- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437

rstp

Syntax

```
rstp {
  disable;
  bpdu-block-on-edge;
  bridge-priority priority;
  forward-delay seconds;
  hello-time seconds;
  interface (all | interface-name) {
    bpdu-timeout-action {
      block;
      alarm;
    }
    disable;
    cost cost;
    edge;
    mode mode;
    no-root-port;
    priority priority;
  }
  max-age seconds;
  traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <(world-readable | no-world-readable)>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure Rapid Spanning Tree Protocol (RSTP). RSTP is defined in the IEEE 802.1D-2004 specification and is used to prevent loops in Layer 2 networks, providing shorter convergence times than those provided with basic STP.

The statements are explained separately.

Default RSTP is enabled on all Ethernet switching interfaces.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding RSTP on page 53
- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437

stp

Syntax

```
stp {
  disable;
  bridge-priority priority;
  forward-delay seconds;
  hello-time seconds;
  interface (all | interface-name) {
    disable;
    bpdu-timeout-action {
      block;
      alarm;
    }
    cost  cost;
    edge;
    mode mode;
    no-root-port;
    priority priority;
  }
  max-age seconds;
  traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <(world-readable | no-world-readable)>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description When you explicitly configure STP, a switch uses the IEEE 802.1D 2004 specification, force version 0. This configuration runs a version of RSTP that is compatible with the classic, basic STP (defined in the IEEE 802.1D 1998 specification).

The remaining statements are explained separately.

Default STP is disabled; by default, RSTP is enabled on all Ethernet switching ports.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring BPDU Protection on STP Interfaces to Prevent STP Miscalculations on page 161
- Configuring STP on page 259
- Overview of Spanning-Tree Protocols on page 51
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

traceoptions (STP)

Syntax

```
traceoptions {
  file name <replace> <size size> <files number> <no-stamp>
    <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit protocols mstp],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]
[edit protocols layer2-control]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

NOTE: **traceoptions is not supported on QFabric systems.**

Description

Set STP protocol-level tracing options.

Default

Traceoptions is disabled.

Options

disable—(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *name* —Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place STP tracing output in the file **/var/log/stp-log**.

files *number* —(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the size option.

Range: 2 through 1000 files

Default: 1 trace file only

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. These are the STP-specific tracing options:

- **all**—Trace all operations.
- **all-failures**—Trace all failure conditions.
- **bpdu**—Trace BPDU reception and transmission.

- **bridge-detection-state-machine**—Trace the bridge detection state machine.
- **events**—Trace events of the protocol state machine.
- **port-information-state-machine**—Trace the port information state machine.
- **port-migration-state-machine**—Trace the port migration state machine.
- **port-receive-state-machine**—Trace the port receive state machine.
- **port-role-select-state-machine**—Trace the port role selection state machine.
- **port-role-transit-state-machine**—Trace the port role transit state machine.
- **port-transmit-state-machine**—Trace the port transmit state machine.
- **port-state-transit-state-machine**—Trace the port state transit state machine.
- **ppmd**—Trace the state and events for the ppmd process.
- **state-machine-variables**—Trace when the state machine variables change.
- **timers**—Trace protocol timers.
- **topology-change-state-machine**—Trace the topology change state machine.

The following are the global tracing options:

- **all**—All tracing operations.
- **config-internal**—Trace configuration internals.
- **general**—Trace general events.
- **normal**—All normal events.

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **parse**—Trace configuration parsing.
- **policy**—Trace policy operations and actions.
- **regex-parse**—Trace regular-expression parsing.
- **route**—Trace routing table changes.
- **state**—Trace state transitions.
- **task**—Trace protocol task processing.
- **timer**—Trace protocol task timer processing.

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Prevent any user from reading the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size ***size***—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0.** When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Example: Configuring Faster Convergence and Improving Network Stability with RSTP on page 165
- Understanding RSTP on page 53
- Understanding MSTP on page 52
- Overview of Spanning-Tree Protocols on page 51
- Understanding VSTP on page 54
- **show spanning-tree bridge on page 432**
- **show spanning-tree interface on page 437**

vlan (STP)

Syntax

```
vlan (vlan-id | vlan-name) {
  bridge-priority priority;
  forward-delay seconds;
  hello-time seconds;
  interface (all | interface-name) {
    bpdu-timeout-action {
      block;
      alarm;
    }
    cost  cost;
    disable;
    edge;
    mode mode;
    no-root-port;
    priority priority;
  }
  max-age seconds;
  traceoptions {
    file filename <files number > <size size> <no-stamp | world-readable |
      no-world-readable>;
    flag flag;
  }
}
```

Hierarchy Level

[edit protocolsmstpmsti *msti-id*],
[edit protocols vstp]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure the VLANs for a Multiple Spanning Tree Instance (MSTI).

The remaining statements are explained separately.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type ? after vlan or vlans in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

Default

Not enabled.

Options

vlan-id—Numeric VLAN identifier.

vlan-name—Name of the VLAN.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Network Regions for VLANs with MSTP on page 184
- Understanding MSTP on page 52
- Understanding VSTP on page 54

vstp

Syntax

```
vstp {
  disable;
  bpdu-block-on-edge;
  force-version stp;
  vlan (vlan-id | vlan-name) {
    bridge-priority priority;
    forward-delay seconds;
    hello-time seconds;
    interface (all | interface-name) {
      disable;
        bpdu-timeout-action {
          alarm;
          block;
      }
      cost  cost;
      edge;
      mode mode;
      no-root-port;
      priority priority;
    }
    max-age seconds;
    traceoptions {
      file name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure VLAN Spanning Tree Protocol (VSTP). VSTP is used to prevent loops in Layer 2 networks on a per-VLAN basis.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding VSTP on page 54
- show spanning-tree bridge on page 432
- show spanning-tree interface on page 437

CHAPTER 39

Static ARP Configuration Statement

arp (Interfaces)

Syntax arp *ip-address* (mac | multicast-mac) *mac-address* publish;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number*family inetaddress *address*],
[edit logical-systems *logical-system-name* interfaces *interface-name*unit *logical-unit-number* family inetaddress *address*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces only, configure Address Resolution Protocol (ARP) table entries, mapping IP addresses to MAC addresses.

Options ***ip-address***—IP address to map to the MAC address. The IP address specified must be part of the subnet defined in the enclosing **address** statement.

mac *mac-address*—MAC address to map to the IP address. Specify the MAC address as six hexadecimal bytes in one of the following formats: ***nnnn.nnnn.nnnn*** or ***nn:nn:nn:nn:nn:nn***. For example, **0011.2233.4455** or **00:11:22:33:44:55**.

multicast-mac *mac-address*—Multicast MAC address to map to the IP address. Specify the multicast MAC address as six hexadecimal bytes in one of the following formats: ***nnnn.nnnn.nnnn*** or ***nn:nn:nn:nn:nn:nn***. For example, **0011.2233.4455** or **00:11:22:33:44:55**.

publish—(Optional) Have the router or switch reply to ARP requests for the specified IP address. If you omit this option, the router or switch uses the entry to reach the destination but does not reply to ARP requests.

NOTE: **The edit logical-systems hierarchy is not available on QFabric systems.**

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Static ARP Table Entries*
- Configuring Static ARP Entries on page 263

CHAPTER 40

VLAN Configuration Statements

description (VLAN)

Syntax description *text-description*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Provide a textual description for the VLAN. The text has no effect on the operation of the VLAN or switch.

Options ***text-description***—Text to describe the interface. It can contain letters, numbers, and hyphens (-) and can be up to 255 characters long. If the text includes spaces, enclose the entire text in quotation marks.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Understanding Bridging and VLANs on page 5
- show vlans on page 448

drop-threshold

Syntax drop-threshold *number*;

Hierarchy Level [edit vlans *vlan-name* dot1q-tunneling layer2-protocol-tunneling (all | *protocol-name*)]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.

Description Specify the maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the switch begins dropping the Layer 2 PDUs. The drop threshold value must be less than or equal to the shutdown threshold value.

L2PT processing is done by the CPU, and L2PT traffic to the CPU is rate-limited to a maximum of 1000 pps. If traffic is received at a rate faster than this limit, the rate limit causes the traffic to be dropped before it hits the threshold and the dropped packets are not reported in L2PT statistics. This can also occur if you configure a drop threshold that is less than 1000 pps but traffic is received at a faster rate. For example, if you configure a drop threshold of 900 pps and the VLAN receives traffic at rate of 1100 pps, L2PT statistics will show that 100 packets were dropped. The 100 packets dropped because of the rate limit will not be reported. Similarly, if you do not configure a drop threshold and the VLAN receives traffic at rate of 1100 pps, the 100 packets dropped because of the rate limit are not reported.

NOTE: If the drop threshold value is greater than the shutdown threshold value and you try to commit the configuration, the commit operation fails.

You can specify a drop threshold value without specifying a shutdown threshold value.

Default No drop threshold is specified.

Options ***number***—Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the interfaces in a specified VLAN before the switch begins dropping the Layer 2 PDUs.

Range: 1 through 1000

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches*
- *Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*
- *Configuring Layer 2 Protocol Tunneling*
- *shutdown-threshold*

filter (VLANs)

Syntax filter (input | output) *filter-name*;

Hierarchy Level [edit vlans *vlan-name*]
[edit vlans *vlan-name* forwarding-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Apply a firewall filter to traffic ingressing or egressing a VLAN.

Default All incoming traffic is accepted unmodified to a VLAN, and all outgoing traffic is sent unmodified from a VLAN.

Options ***filter-name***—Name of a firewall filter defined at the **[edit firewall family *family-name* filter]** hierarchy level.

input—Apply a firewall filter to VLAN ingress traffic.

output—Apply a firewall filter to VLAN egress traffic.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Firewall Filters*
- *Overview of Firewall Filters*

interface (VLANs)

Syntax

```
interface interface-name {
  mapping (native (push | swap) | tag (push | swap));
}
```

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For a specific VLAN, configure an interface.

Options ***interface-name***—Name of the interface.

The remaining statement is explained separately.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Configuring VLANs on page 220
- *Understanding Bridging*

l3-interface (VLAN)

Syntax l3-interface (vlan.*logical-interface-number* | irb.*logical-interface-number*);

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
irb option introduced in Junos OS Release 13.2 for the QFX Series.

Description Associate a Layer 3 interface with the VLAN. Configure Layer 3 interfaces on trunk ports to allow the interface to transfer traffic between VLANs. Traffic between VLANs must be routed, which requires a common Layer 3 interface.

Default No Layer 3 (routing) interface is associated with the VLAN.

Options vlan.***logical-interface-number***—Number of the logical interface. Use the **unit** number that you used when you created the **vlan** interface with a **set interfaces vlan unit** statement.

NOTE: Use this statement with versions of Junos OS that do not support Enhanced Layer 2 Software (ELS).

irb.***logical-interface-number***—Logical interface defined with a **set interfaces irb** statement.

NOTE: Use this statement with versions of Junos OS that support Enhanced Layer 2 Software (ELS).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- show ethernet-switching interfaces on page 388
- show vlans on page 448

members

Syntax members [(all | *names* | *vlan-ids*)];

Hierarchy Level [edit interfaces *interface-name* unit 0 family ethernet-switching vlan]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description For trunk interfaces, configure the VLANs for which the interface can carry traffic.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type ? after vlan or vlans in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

Options **all**—Specify that this trunk interface be a member of all the VLANs that are configured on this switch. When a new VLAN is configured on the switch, this trunk interface automatically becomes a member of the VLAN.

NOTE: Each VLAN that is configured must have a specified VLAN ID when you attempt to commit the configuration; otherwise, the configuration commit fails. Also, all cannot be the name of a VLAN on the switch.

names—Names of one or more VLANs.

vlan-ids—Numeric identifiers of one or more VLANs.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Understanding Bridging and VLANs on page 5
- **show ethernet-switching interfaces on page 388**
- **show vlans on page 448**

native-vlan-id

Syntax native-vlan-id *vlan-id*;

Hierarchy Level For platforms without ELS:

[edit interfaces *interface-name* unit 0 family ethernet-switching],

For platforms with ELS:

[edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the VLAN identifier to associate with untagged packets received on the interface. The logical interface on which untagged packets are received must be configured with the same VLAN ID as the native VLAN ID configured on the physical interface. To configure the logical interface, include the **vlan-id** statement (matching the **native-vlan-id** statement on the physical interface) at the **[edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level**.

When the **native-vlan-id** statement is combined with the *interface-mode* statement, untagged packets are accepted and forwarded within the bridge domain or VLAN that is configured with the matching VLAN ID.

When the **native-vlan-id** statement is combined with the *flexible-vlan-tagging* statement, untagged packets are accepted on the interfaces that are configured for Q-in-Q tunneling.

.

Options ***vlan-id***—Numeric identifier of the VLAN.

Range: 1 through 4094

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Junos OS Network Interfaces Configuration Guide*
-
- show ethernet-switching interfaces on page 388
- show vlans on page 448

port-mode

Syntax port-mode (access | tagged-access | trunk);

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

NOTE: This statement does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *interface-mode*. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

Configure whether an interface on the switch operates in access, tagged access, or trunk mode.

Default All switch interfaces are in access mode.

Options

access—Have the interface operate in access mode. In this mode, the interface can be in a single VLAN only. Access interfaces typically connect to network devices, such as PCs, printers, IP telephones, and IP cameras.

tagged-access—Have the interface operate in tagged-access mode. In this mode, the interface can be in multiple VLANs. Tagged access interfaces typically connect to network devices, such as PCs, printers, IP telephones, and IP cameras.

trunk—Have the interface operate in trunk mode. In this mode, the interface can be in multiple VLANs and can multiplex traffic between different VLANs. Trunk interfaces typically connect to other switches and to routers on the LAN.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring Reflective Relay on page 255
- Example: Configuring Reflective Relay for Use with VEPA Technology on page 155

vlan (Ethernet)

Syntax

```
vlan {
  members [ (all | names | vlan-ids) ];
}
```

Hierarchy Level

[edit interfaces ge-*chassis/slot/port* unit *logical-unit-number* ethernet-switching],
[edit interfaces xe-*chassis/slot/port* unit *logical-unit-number* ethernet-switching]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

For Gigabit Ethernet and aggregated Ethernet interfaces, assign an 802.1Q VLAN tag ID to a logical interface.

The statements are explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Junos OS Network Interfaces Configuration Guide*

vlan (Unknown Unicast)

Syntax

```
vlan (all | vlan-name) {
  interface interface-name;
}
```

Hierarchy Level [edit ethernet-switching-options unknown-unicast-forwarding]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Specify a VLAN from which unknown unicast packets will be forwarded, or specify that the packets should be forwarded from *all* VLANS. Unknown unicast packets are forwarded from a VLAN to a specific trunk interface.

The remaining statement is explained separately.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type ? after vlan or vlans in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range.

Options all—All VLANs.

vlan-name—Name of a VLAN.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Unknown Unicast Forwarding*
- *Verifying That Unknown Unicast Packets Are Forwarded to a Trunk Interface*
- *Understanding Unknown Unicast Forwarding*
- show ethernet-switching table on page 408
- show vlans on page 448

vlan-id (VLANs)

Syntax vlan-id *number*;

Hierarchy Level For platforms without ELS:

[edit vlans *vlan-name* vlan-range]

For platforms without ELS and with ELS:

[edit vlans *vlan-name*]

For ELS platforms only:

[edit interfaces *interface-name* unit *number*]
[edit vlans *vlan-name* vlan-id-list]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure an 802.1Q tag to apply to all traffic that originates on the VLAN.

Default On a QFX3500 and QFX3500 switch, if you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**. The number zero is reserved for priority tagging and the number 4093 is also reserved.

On a QFX5100 switch, if you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**. The number zero is reserved for priority tagging and the number 4093 is also reserved.

NOTE: You can only create up 4090 VLANs on a QFX5100 switch. If you create more than 4090 VLANs, the interfaces associated with the extra VLANs are not displayed in the **show vlans** command output. For example, if you create 4094 VLANs, the extra VLANs will not have interfaces associated with the VLANs. The order in which you configure the extra VLANs determines which interfaces are missing from the **show vlans** command output.

Options *number* —VLAN tag identifier.

Range: 0 through 4093.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Understanding Bridging*

vlan-range

Syntax vlan-range *vlan-id-low-vlan-id-high*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure multiple VLANs. Each VLAN is assigned a VLAN ID number from the range.

Default None.

Options ***vlan-id-low-vlan-id-high***—Specify the first and last VLAN ID number for the group of VLANs.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Configuring VLANs on page 220
- Configuring IRB Interfaces on page 257
- *Understanding Bridging*

vlans

Syntax

```
vlans {
  vlan-name {
    description text-description;
    dot1q-tunneling {
      customer-vlans (id | range);
    }
    filter input filter-name;
    filter output filter-name;
    interface interface-name {
      isolated;
      mapping (policy | tag push | native push);
      promiscuous;
    }
    isolation-vlan-id;
    l3-interface vlan.logical-interface-number;
    mac-limit number;
    mac-table-aging-time seconds;
    no-local-switching;
    no-mac-learning;
    primary-vlan vlan-name;
    pvlan extend-secondary-vlan-id vlan-id;
    vlan-id number;
    vlan-range vlan-id-low-vlan-id-high;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statements for private VLANs and Q-in-Q tunneling introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure VLAN properties on the QFX Series.

Default If you use the default factory configuration, all switch interfaces become part of the VLAN **default**.

Options ***vlan-name***—Name of the VLAN. The name can contain letters, numbers, hyphens (-), and periods (.) and can be up to 255 characters long.

The remaining statements are described separately.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Configuring VLANs on page 220
- Configuring Q-in-Q Tunneling on page 253
- Creating a Series of Tagged VLANs on page 222

- Configuring IRB Interfaces on page 257
- Creating a Private VLAN on a Single Switch on page 247
- *Understanding Bridging*

vlan-tagging

Syntax vlan-tagging;

Hierarchy Level [edit interfaces *interface-name*]
[edit interfaces interface-range *interface-range-name*]

Release Information Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Enable VLAN tagging. The platform receives and forwards single-tag frames with 802.1Q VLAN tags.

Default VLAN tagging is disabled by default.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *vlan-id*
- *Configuring a Layer 3 Logical Interface*

PART 3

Administration

CHAPTER 41

Routine Monitoring

Verifying That MAC Notification Is Working Properly

Purpose Verify that MAC notification is enabled or disabled, and that the MAC notification interval is set to the specified value.

Action To verify that MAC notification is enabled or disabled and also to verify the MAC notification interval setting.

```
user@switch> show ethernet-switching mac-notification
Notification Status: Enabled
Notification Interval: 30
```

Meaning The output in the **Notification Status** field shows that MAC notification is enabled. The output in the **Notification Status** field would display **Disabled** if MAC notification was disabled.

The **Notification Interval** field output shows that the MAC notification interval is set to 30 seconds.

Related Documentation

- Configuring MAC Notification on page 239

Verifying That a Series of Tagged VLANs Has Been Created

Purpose Verify that a series of tagged VLANs has been created on the switch.

Action

1. Display the VLANs in the ascending order of their VLAN ID:

```
user@switch> show vlans sort-by tag

Name             Tag         Interfaces
```

```
__employee_120__  120
                                xe-0/0/22.0*
__employee_121__  121
                                xe-0/0/22.0*
__employee_122__  122
                                xe-0/0/22.0*
__employee_123__  123
                                xe-0/0/22.0*
__employee_124__  124
                                xe-0/0/22.0*
__employee_125__  125
                                xe-0/0/22.0*
__employee_126__  126
                                xe-0/0/22.0*
__employee_127__  127
                                xe-0/0/22.0*
__employee_128__  128
                                xe-0/0/22.0*
__employee_129__  129
                                xe-0/0/22.0*
__employee_130__  130
                                xe-0/0/22.0*
```

2. Display the VLANs by the alphabetical order of the VLAN name:

```
user@switch>  show vlans sort-by name

Name             Tag   Interfaces

__employee_120__ 120
                       xe-0/0/22.0*
__employee_121__ 121
                       xe-0/0/22.0*
__employee_122__ 122
                       xe-0/0/22.0*
__employee_123__ 123
                       xe-0/0/22.0*
__employee_124__ 124
                       xe-0/0/22.0*
__employee_125__ 125
                       xe-0/0/22.0*
__employee_126__ 126
                       xe-0/0/22.0*
__employee_127__ 127
                       xe-0/0/22.0*
__employee_128__ 128
                       xe-0/0/22.0*
__employee_129__ 129
                       xe-0/0/22.0*
__employee_130__ 130
                       xe-0/0/22.0*
```

3. Display the VLANs by specifying the VLAN range name (here, the VLAN range name is **employee**):

```
user@switch>  show vlans employee

Name             Tag   Interfaces

__employee_120__ 120
                       xe-0/0/22.0*
__employee_121__ 121
```

```
                                xe-0/0/22.0*
__employee_122__ 122
                                xe-0/0/22.0*
__employee_123__ 123
                                xe-0/0/22.0*
__employee_124__ 124
                                xe-0/0/22.0*
__employee_125__ 125
                                xe-0/0/22.0*
__employee_126__ 126
                                xe-0/0/22.0*
__employee_127__ 127
                                xe-0/0/22.0*
__employee_128__ 128
                                xe-0/0/22.0*
__employee_129__ 129
                                xe-0/0/22.0*
__employee_130__ 130
                                xe-0/0/22.0*
```

Meaning The sample output shows the VLANs configured on the switch. The series of tagged VLANs is displayed: **__employee_120__** through **__employee_130__**. Each of the tagged VLANs is configured on the trunk interface **xe-0/0/22.0**. The asterisk (*) next to the interface name indicates that the interface is **UP**.

When a series of VLANs is created using the **vlan-range** statement, the VLAN names are preceded and followed by a double underscore.

Related Documentation

- Creating a Series of Tagged VLANs on page 222
- *Creating a Series of Tagged VLANs*

Verifying That Q-in-Q Tunneling Is Working

Purpose After creating a Q-in-Q VLAN, verify that it is set up properly.

Action

1. Use the **show configuration vlans** command to determine if you successfully created the primary and secondary VLAN configurations:

```
user@switch> show configuration vlans
svlan {
   vlan-id 300;
   dot1q-tunneling {
      customer-vlans [ 101–200 ];
   }
}
```

2. Use the **show vlans** command to view VLAN information and link status:

```
user@switch> show vlans s-vlan-name extensive
VLAN: svlan, Created at: Thu Oct 23 16:53:20 2008
802.1Q Tag: 300, Internal index: 2, Admin State: Enabled, Origin: Static
Dot1q Tunneling Status: Enabled
Customer VLAN ranges:
                     101–200
Protocol: Port Mode
Number of interfaces: Tagged 1 (Active = 0), Untagged  1 (Active = 0)
```

```
        xe-0/0/1, tagged, trunk
        xe-0/0/2, untagged, access
```

Meaning The output confirms that Q-in-Q tunnling is enabled and that the VLAN is tagged, and lists the customer VLANs that are associated with the tagged VLAN.

Related Documentation

- *Configuring Q-in-Q Tunneling (CLI Procedure)*
- *Example: Setting Up Q-in-Q Tunneling on EX Series Switches*

Verifying That a Private VLAN Is Working

Purpose After creating and configuring private VLANs (PVLANs), verify that they are set up properly.

Action

1. To determine whether you successfully created the primary and secondary VLAN configurations:

   - For a PVLAN on a single switch, use the **show configuration vlans** command:

     ```
     user@switch> show configuration vlans
     community1 {
         interface {
             interface a;
             interface b;
         }
         primary-vlan pvlan;
     }
     community2 {
         interface {
             interface d;
             interface e;
         }
         primary-vlan pvlan;
     }
     pvlan {
         vlan-id 1000;
         interface {
             isolated1;
             isolated2;
             trunk1;
             trunk2;
         }
         no-local-switching;
     }
     ```

   - For a PVLAN spanning multiple switches, use the ***show vlans* extensive** command:

     ```
     user@switch> show vlans extensive
     VLAN: COM1, Created at: Tue May 11 18:16:05 2010
     802.1Q Tag: 100, Internal index: 3, Admin State: Enabled, Origin: Static
     Private VLAN Mode: Community, Primary VLAN: primary
     Protocol: Port Mode, Mac aging time: 300 seconds
     Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
           ge-0/0/20.0*, tagged, trunk
           ge-0/0/22.0*, tagged, trunk, pvlan-trunk
           ge-0/0/23.0*, tagged, trunk, pvlan-trunk
           ge-0/0/7.0*, untagged, access
     ```

```
VLAN: __pvlan_primary_ge-0/0/0.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 5, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access


VLAN: __pvlan_primary_ge-0/0/2.0__, Created at: Tue May 11 18:16:05 2010
Internal index: 6, Admin State: Enabled, Origin: Static
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/2.0, untagged, access


VLAN: __pvlan_primary_isiv__, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  0 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk


VLAN: community2, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 20, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  2 (Active = 2)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, untagged, access
      ge-1/0/6.0*, untagged, access


VLAN: primary, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 10, Internal index: 2, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  5 (Active = 4)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access
      ge-0/0/1.0*, untagged, access
      ge-0/0/2.0, untagged, access
      ge-0/0/7.0*, untagged, access
      ge-1/0/6.0*, untagged, access

Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_primary_ge-0/0/0.0__
```

```
    __pvlan_primary_ge-0/0/2.0__
  Community VLANs :
    COM1
    community2
  Inter-switch-isolated VLAN :
    __pvlan_primary_isiv__
```

2. Use the **show vlans extensive** command to view VLAN information and link status for a PVLAN on a single switch or for a PVLAN spanning multiple switches.

   - For a PVLAN on a single switch:

     ```
     user@switch> show vlans pvlan extensive
     VLAN: pvlan, Created at: time
     802.1Q Tag: vlan-id, Internal index: index-number, Admin State: Enabled,
     Origin: Static
     Private VLAN Mode: Primary
     Protocol: Port Mode
     Number of interfaces: Tagged 2 (Active = 0), Untagged  6 (Active = 0)
           trunk1, tagged, trunk
           interface a, untagged, access
           interface b, untagged, access
           interface c, untagged, access
           interface d, untagged, access
           interface e, untagged, access
           interface f, untagged, access
           trunk2, tagged, trunk
     Secondary VLANs: Isolated 2, Community  2
       Isolated VLANs :
           __pvlan_pvlan_isolated1__
           __pvlan_pvlan_isolated2__
       Community VLANs :
           community1
           community2
     ```

   - For a PVLAN spanning multiple switches:

     ```
     user@switch> show vlans extensive
     VLAN: COM1, Created at: Tue May 11 18:16:05 2010
     802.1Q Tag: 100, Internal index: 3, Admin State: Enabled, Origin: Static
     Private VLAN Mode: Community, Primary VLAN: primary
     Protocol: Port Mode, Mac aging time: 300 seconds
     Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
           ge-0/0/20.0*, tagged, trunk
           ge-0/0/22.0*, tagged, trunk, pvlan-trunk
           ge-0/0/23.0*, tagged, trunk, pvlan-trunk
           ge-0/0/7.0*, untagged, access

     VLAN: __pvlan_primary_ge-0/0/0.0__, Created at: Tue May 11 18:16:05 2010
     Internal index: 5, Admin State: Enabled, Origin: Static
     Private VLAN Mode: Isolated, Primary VLAN: primary
     Protocol: Port Mode, Mac aging time: 300 seconds
     Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 1)
           ge-0/0/20.0*, tagged, trunk
           ge-0/0/22.0*, tagged, trunk, pvlan-trunk
           ge-0/0/23.0*, tagged, trunk, pvlan-trunk
           ge-0/0/0.0*, untagged, access


     VLAN: __pvlan_primary_ge-0/0/2.0__, Created at: Tue May 11 18:16:05 2010
     Internal index: 6, Admin State: Enabled, Origin: Static
     ```

```
Private VLAN Mode: Isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  1 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/2.0, untagged, access


VLAN: __pvlan_primary_isiv__, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 50, Internal index: 7, Admin State: Enabled, Origin: Static
Private VLAN Mode: Inter-switch-isolated, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  0 (Active = 0)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk


VLAN: community2, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 20, Internal index: 8, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  2 (Active = 2)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/1.0*, untagged, access
      ge-1/0/6.0*, untagged, access


VLAN: primary, Created at: Tue May 11 18:16:05 2010
802.1Q Tag: 10, Internal index: 2, Admin State: Enabled, Origin: Static
Private VLAN Mode: Primary
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 3 (Active = 3), Untagged  5 (Active = 4)
      ge-0/0/20.0*, tagged, trunk
      ge-0/0/22.0*, tagged, trunk, pvlan-trunk
      ge-0/0/23.0*, tagged, trunk, pvlan-trunk
      ge-0/0/0.0*, untagged, access
      ge-0/0/1.0*, untagged, access
      ge-0/0/2.0, untagged, access
      ge-0/0/7.0*, untagged, access
      ge-1/0/6.0*, untagged, access

Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_primary_ge-0/0/0.0__
      __pvlan_primary_ge-0/0/2.0__
  Community VLANs :
      COM1
      community2
  Inter-switch-isolated VLAN :
      __pvlan_primary_isiv__
```

3. Use the **show ethernet-switching table** command to view logs for MAC learning on the VLANs:

   ```
   user@switch> show ethernet-switching table
   ```

```
Ethernet-switching table: 8 entries, 1 learned
  VLAN              MAC address       Type         Age Interfaces
  default           *                 Flood          - All-members
  pvlan             *                 Flood          - All-members
  pvlan             MAC1              Replicated     - interface a
  pvlan             MAC2              Replicated     - interface c
  pvlan             MAC3              Replicated     - isolated2
  pvlan             MAC4              Learn          0 trunk1
  __pvlan_pvlan_isolated1__ *         Flood          - All-members
  __pvlan_pvlan_isolated1__ MAC4      Replicated     - trunk1
  __pvlan_pvlan_isolated2__ *         Flood          - All-members
  __pvlan_pvlan_isolated2__ MAC3      Learn          0 isolated2
  __pvlan_pvlan_isolated2__ MAC4      Replicated     - trunk1
  community1        *                 Flood          - All-members
  community1        MAC1              Learn          0 interface a
  community1        MAC4              Replicated     - trunk1
  community2        *                 Flood          - All-members
  community2        MAC2              Learn          0 interface c
  community2        MAC4              Replicated     - trunk1
```

NOTE: If you have configured a PVLAN spanning multiple switches, you can use the same command on all the switches to check the logs for MAC learning on those switches.

Meaning In the output displays for a PVLAN on a single switch, you can see that the primary VLAN contains two community domains (**community1** and **community2**), two isolated ports, and two trunk ports. The PVLAN on a single switch has only one tag (**1000**), which is for the primary VLAN.

The PVLAN that spans multiple switches contains multiple tags:

- The community domain **COM1** is identified with tag **100**.
- The community domain **community2** is identified with tag **20**.

- The interswitch isolated domain is identified with tag **50**.
- The primary VLAN **primary** is identified with tag **10**.

Also, for the PVLAN that spans multiple switches, the trunk interfaces are identified as **pvlan-trunk**.

Related Documentation

- *Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)*
- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*
- Creating a Private VLAN on a Single Switch on page 247
- Creating a Private VLAN Spanning Multiple Switches on page 249

Verifying That Proxy ARP Is Working Correctly

Purpose Verify that the switch is sending proxy ARP messages.

Action List the system statistics for ARP:

```
user@switch> show system statistics arp
arp:
        90060 datagrams received
        34 ARP requests received
        610 ARP replies received
        2 resolution request  received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 unrestricted proxy requests not proxied
        0 restricted proxy requests not proxied
        0 datagrams with bogus interface
        0 datagrams with incorrect length
        0 datagrams for non-IP protocol
        0 datagrams with unsupported op code
        0 datagrams with bad protocol address length
        0 datagrams with bad hardware address length
        0 datagrams with multicast source address
        0 datagrams with multicast target address
        0 datagrams with my own hardware address
        0 datagrams for an address not on the interface
        0 datagrams with a broadcast source address
        294 datagrams with source address duplicate to mine
        89113 datagrams which were not for me
        0 packets discarded waiting for resolution
        0 packets sent after waiting for resolution
        309 ARP requests sent
        35 ARP replies sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
```

```
    0 replies from unnumbered interface with non-subnetted donor
```

Meaning The statistics show that two proxy ARP requests were received. The **unrestricted proxy requests not proxied** and **restricted proxy requests not proxied** fields indicate that all the unproxied ARP requests received have been proxied by the switch.

Related Documentation

- Configuring Proxy ARP on page 251
- *Configuring Proxy ARP (CLI Procedure)*

Verifying That MVRP Is Working Correctly

Purpose After configuring your switch to participate in MVRP, verify that the configuration is properly set and that MVRP messages are being sent and received on your switch.

Action

1. Confirm that MVRP is enabled on your switch.

```
user@switch> show mvrp

Global MVRP configuration
  MVRP status               : Enabled
  MVRP dynamic vlan creation: Enabled
  MVRP Timers (ms):
   Interface        Join   Leave   LeaveAll
   --------------   ----   -----   --------
   all              200    600     10000
   xe-0/1/1.0       200    600     10000

  Interface based configuration:
   Interface        Status     Registration   Dynamic VLAN Creation
   --------------   --------   ------------   ---------------------
   all          Disabled   Fixed          Enabled
   xe-0/1/1.0       Enabled    Normal         Enabled
```

2. Confirm that MVRP messages are being sent and received on your switch.

```
user@switch> show mvrp statistics interface xe-0/1/1.0
MVRP statistics
  MRPDU received                : 3342
  Invalid PDU received          : 0
  New received                  : 2
  Join Empty received           : 1116
  Join In received              : 2219
  Empty received                : 2
  In received                   : 2
  Leave received                : 1
  LeaveAll received             : 1117
  MRPDU transmitted             : 3280
  MRPDU transmit failures       : 0
  New transmitted               : 0
  Join Empty transmitted        : 1114
  Join In transmitted           : 2163
  Empty transmitted             : 1
  In transmitted                : 1
  Leave transmitted             : 1
  LeaveAll transmitted          : 1111
```

Meaning The output of **show mvrp** shows that interface **xe-0/1/1.0** is enabled for MVRP participation as shown in the status in the **Interface based configuration** field.

The output for **show mvrp statistics interface xe-0/1/1.0** confirms that MVRP messages are being transmitted and received on the interface.

NOTE: You can identify an MVRP compatibility issue on EX Series switches by looking at the output from this command. If *Join Empty received* and *Join In received* incorrectly display zero, even though the value for *MRPDU received* has been increased, you are probably running different versions of Junos OS, including Release 11.3, on the switches in this network. Another indication that MVRP is having a version problem is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the earlier release version. To remedy these problems, see *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*.

Related Documentation

- *Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches*
- Example: Configuring Automatic VLAN Administration Using MVRP on page 71
- *Configuring Multiple VLAN Registration Protocol (MVRP) (CLI Procedure)*

CHAPTER 42

Monitoring Commands

- clear bpdu-error
- clear ethernet-switching layer2-protocol-tunneling error
- clear ethernet-switching layer2-protocol-tunneling statistics
- clear ethernet-switching table
- clear spanning-tree statistics
- show ethernet-switching interfaces
- show ethernet-switching layer2-protocol-tunneling interface
- show ethernet-switching layer2-protocol-tunneling statistics
- show ethernet-switching layer2-protocol-tunneling vlan
- show ethernet-switching mac-learning-log
- show ethernet-switching mac-notification
- show ethernet-switching statistics aging
- show ethernet-switching statistics mac-learning
- show ethernet-switching table
- show interfaces xe
- show spanning-tree bridge
- show spanning-tree interface
- show spanning-tree mstp configuration
- show spanning-tree statistics
- show system statistics arp
- show vlans

clear bpdu-error

Syntax clear bpdu-error interface *interface-name*

Release Information Command introduced in Junos OS Release 9.1 for EX Series switches. Command updated in Junos OS Release 11.1 for EX Series switches—a BPDU error shuts down the interface and this command brings the interface back up.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Clear bridge protocol data unit (BPDU) errors from an interface and bring up the interface.

Options ***interface-name***—Clear BPDU errors on the specified interface.

Required Privilege Level clear

Related Documentation

- show spanning-tree interface on page 437
- *Understanding BPDU Protection for STP, RSTP, and MSTP on EX Series Switches*
- Understanding BPDU Protection for STP, RSTP, and MSTP on page 55

List of Sample Output clear bpdu-error interface on page 382

Sample Output

clear bpdu-error interface

```
user@switch> clear bpdu-error interface xe-0/0/1.0
```

clear ethernet-switching layer2-protocol-tunneling error

Syntax clear ethernet-switching layer2-protocol-tunneling error
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Clear Layer 2 protocol tunneling (L2PT) errors on one or more interfaces. If an interface has been disabled because the amount of Layer 2 protocol traffic exceeded the shutdown threshold or because the switch has detected an error in the network topology or configuration, use this command to reenable the interface.

Options **none**—Clears L2PT errors on all interfaces.

interface *interface-name*—(Optional) Clear L2PT errors on the specified interface.

Required Privilege Level view

Related Documentation
- *Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches*
- *Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*
- *Configuring Layer 2 Protocol Tunneling*

List of Sample Output clear ethernet-switching layer2-protocol-tunneling error on page 383
clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0 on page 383

Sample Output

clear ethernet-switching layer2-protocol-tunneling error

```
user@switch> clear ethernet-switching layer2-protocol-tunneling error
```

clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0

```
user@switch> clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0
```

clear ethernet-switching layer2-protocol-tunneling statistics

Syntax clear ethernet-switching layer2-protocol-tunneling statistics
<interface *interface-name*>
<vlan *vlan-name*>

Release Information Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Clear Layer 2 protocol tunneling (L2PT) statistics on one or more interfaces or VLANs.

Options **none**—Clear L2PT statistics on all interfaces and VLANs.

interface *interface-name*—(Optional) Clear L2PT statistics on the specified interface.

vlan *vlan-name*—(Optional) Clear L2PT statistics on the specified VLAN.

Required Privilege Level view

Related Documentation

- show ethernet-switching layer2-protocol-tunneling statistics on page 394
- *Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches*
- *Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*
- *Configuring Layer 2 Protocol Tunneling*

List of Sample Output

Sample Output

clear ethernet-switching layer2-protocol-tunneling statistics

```
user@switch> clear ethernet-switching layer2-protocol-tunneling statistics
```

clear ethernet-switching layer2-protocol-tunneling error interface ge-0/1/1.0

```
user@switch> clear ethernet-switching layer2-protocol-tunneling statistics interface xe-0/1/1.0
```

clear ethernet-switching layer2-protocol-tunneling error vlan v2

```
user@switch> clear ethernet-switching layer2-protocol-tunneling statistics vlan v2
```

clear ethernet-switching table

Syntax

clear ethernet-switching table
<interface *interface-name*>
<mac *mac-address*>
<management-vlan>
<persistent-mac <*interface* | *mac-address*>>
<vlan *vlan-name*>

Syntax (QFX Series)

clear ethernet-switching table
<interface *interface-name*>
<mac *mac-address*>
<persistent-mac <*interface* | *mac-address*>>
<vlan *vlan-name*>

Release Information

Command introduced in Junos OS Release 9.3 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

NOTE: On a QFabric system, using this command on an FCoE-enabled VLAN when FCoE sessions are active can cause traffic flooding and FCoE traffic drop. The FCoE sessions are not terminated and the traffic reconverges after a short period of time.

Clear learned entries, which are media access control (MAC) addresses, in the Ethernet switching table (also called the forwarding database table).

Options

none—Clear learned entries in the Ethernet switching table, except for persistent MAC addresses.

interface ***interface-name***—(Optional) Clear all learned MAC addresses for the specified interface from the Ethernet switching table.

mac ***mac-address***—(Optional) Clear the specified learned MAC address from the Ethernet switching table.

management-vlan—(Optional) Clear all MAC addresses learned for the management VLAN from the Ethernet switching table. Note that you do not specify a VLAN name because only one management VLAN exists.

persistent-mac <*interface* | *mac-address*>—(Optional) Clear all MAC addresses, including persistent MAC addresses. Use the **interface** option to clear all MAC addresses on an interface, or use the **mac-address** option to clear all entries for a specific MAC address.

Use this command whenever you move a device in your network that has a persistent MAC address on the switch. If you move the device to another port on the switch and do not clear the persistent MAC address from the original port it was learned on, then the new port will not learn the MAC address and the device will not be able to connect. If the original port is down when you move the device, then the new port

will learn the MAC address and the device can connect—however, unless you cleared the MAC address on the original port, when the port comes back up, the system reinstalls the persistent MAC address in the forwarding table for that port. If this occurs, the address is removed from the new port and the device loses connectivity.

vlan *vlan-name*—(Optional) Clear all MAC addresses learned for the specified VLAN from the Ethernet switching table.

Required Privilege Level view

Related Documentation
- *show ethernet-switching table*
- show ethernet-switching table on page 408
- *Verifying That Persistent MAC Learning Is Working Correctly*

List of Sample Output clear ethernet-switching table on page 386

Output Fields This command produces no output.

Sample Output

clear ethernet-switching table

```
user@switch> clear ethernet-switching table
```


clear spanning-tree statistics

List of Syntax
Syntax on page 387
Syntax (EX Series Switches and the QFX Series) on page 387

Syntax
clear spanning-tree statistics
<interface *interface-name*>
<logical-system *logical-system-name*>

Syntax (EX Series Switches and the QFX Series)
clear spanning-tree statistics
<interface *interface-name*>

Release Information
Command introduced in Junos OS Release 8.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description
Clear Spanning Tree Protocol statistics.

Options
none—Reset STP counters for all interfaces for all routing instances.

interface *interface-name*—(Optional) Clear STP statistics for the specified interface only.

logical-system *logical-system-name*—(Optional) Clear STP statistics on a particular logical system.

NOTE: The logical-system option is not available on QFabric systems.

Required Privilege Level
clear

Related Documentation
- show spanning-tree statistics on page 445

List of Sample Output
clear stp statistics on page 387

Sample Output

clear stp statistics

```
user@host> clear stp statistics
```

show ethernet-switching interfaces

Syntax show ethernet-switching interfaces
<brief | detail | summary>
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display information about switched Ethernet interfaces.

Options **none**—(Optional) Display brief information for Ethernet-switching interfaces.

brief | detail | summary—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display Ethernet-switching information for a specific interface.

Required Privilege Level view

Related Documentation

- Troubleshooting Ethernet Switching on page 459Understanding Bridging and VLANs on page 5
- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Understanding FCoE*
- *Interfaces Overview*

List of Sample Output

show ethernet-switching interfaces on page 389
show ethernet-switching interfaces summary on page 390
show ethernet-switching interfaces brief on page 390
show ethernet-switching interfaces detail on page 390
show ethernet-switching interfaces interface-name on page 391

Output Fields Table 34 on page 388 lists the output fields for the **show ethernet-switching interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 34: show ethernet-switching interfaces Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a switching interface.	All levels
State	Interface state. Values are **up** or **down**.	none, **brief**, **detail**, **summary**
VLAN members	Name of a VLAN.	none, **brief**, **detail**, **summary**

Table 34: show ethernet-switching interfaces Output Fields *(continued)*

Field Name	Field Description	Level of Output
Blocking	Forwarding state of the interface: • **blocked**—Traffic is not being forwarded on the interface. • **unblocked**—Traffic is forwarded on the interface. • **MAC limit exceeded**—The interface is temporarily disabled because of a MAC limiting error. The disabled interface is automatically restored to service when the disable timeout expires. • **MAC move limit exceeded**—The interface is temporarily disabled because of a MAC move limiting error. The disabled interface is automatically restored to service when the disable timeout expires. • **Storm control in effect** —The interface is temporarily disabled because of a storm control error. The disabled interface is automatically restored to service when the disable timeout expires. • **Storm control shutdown in effect** —The interface is temporarily disabled because of a storm control shutdown error. The disabled interface is automatically restored to service when the disable timeout expires.	none, **brief**, **detail**, **summary**
Index	VLAN index internal to Junos OS software.	**detail**
untagged | tagged	Specifies whether the interface forwards IEEE802.1Q-tagged or untagged traffic.	**detail**

Sample Output

show ethernet-switching interfaces

```
user@switch> show ethernet-switching interfaces

Interface   State    VLAN members          Blocking
xe-0/0/0.0  up       T1122                 unblocked
xe-0/0/1.0  down     default               – MAC limit exceeded
xe-0/0/2.0  down     default               – MAC move limit exceeded
xe-0/0/3.0  down     default               – Storm control in effect
xe-0/0/4.0  down     default               unblocked
xe-0/0/5.0  down     default               unblocked
xe-0/0/6.0  down     default               unblocked
xe-0/0/7.0  down     default               unblocked
xe-0/0/8.0  down     default               unblocked
xe-0/0/9.0  up       T111                  unblocked
xe-0/0/10.0 down     default               unblocked
xe-0/0/11.0 down     default               unblocked
xe-0/0/12.0 down     default               unblocked
xe-0/0/13.0 down     default               unblocked
xe-0/0/14.0 down     default               unblocked
xe-0/0/15.0 down     default               unblocked
xe-0/0/16.0 down     default               unblocked
xe-0/0/17.0 down     default               unblocked
xe-0/0/18.0 down     default               unblocked
xe-0/0/19.0 up       T111                  unblocked
xe-0/1/0.0  down     default               unblocked
xe-0/1/1.0  down     default               unblocked
xe-0/1/2.0  down     default               unblocked
xe-0/1/3.0  down     default               unblocked
```

show ethernet-switching interfaces summary

```
user@switch> show ethernet-switching interfaces summary
xe-0/0/0.0
xe-0/0/1.0
xe-0/0/2.0
xe-0/0/3.0
xe-0/0/8.0
xe-0/0/10.0
xe-0/0/11.0
```

show ethernet-switching interfaces brief

```
user@switch> show ethernet-switching interfaces brief
Interface   State    VLAN members           Blocking
xe-0/0/0.0  down     default                unblocked
xe-0/0/1.0  down     employee-vlan          unblocked
xe-0/0/2.0  down     employee-vlan          unblocked
xe-0/0/3.0  down     employee-vlan          unblocked
xe-0/0/8.0  down     employee-vlan          unblocked
xe-0/0/10.0 down     default                unblocked
xe-0/0/11.0 down     employee-vlan          unblocked
```

show ethernet-switching interfaces detail

```
user@switch> show ethernet-switching interfaces detail
Interface: xe-0/0/0.0 Index: 65
  State: down
  VLANs:
    default                untagged     unblocked

Interface: xe-0/0/1.0 Index: 66
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/2.0 Index: 67
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/3.0 Index: 68
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/8.0 Index: 69
  State: down
  VLANs:
    employee-vlan          untagged     unblocked

Interface: xe-0/0/10.0 Index: 70
  State: down
  VLANs:
    default                untagged     unblocked

Interface: xe-0/0/11.0 Index: 71
  State: down
  VLANs:
    employee-vlan          tagged       unblocked
```

show ethernet-switching interfaces interface-name

```
user@switch> show ethernet-switching interfaces xe-0/0/0.0
 Interface   State    VLAN members            Blocking
xe-0/0/0.0  down     default                 unblocked
```

show ethernet-switching layer2-protocol-tunneling interface

Syntax show ethernet-switching-layer2-protocol-tunneling interface <*interface-name*>

Release Information Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Display information about Layer 2 protocol tunneling (L2PT) on interfaces that have been configured for L2PT.

Options **none**—Display L2PT information about all interfaces on which L2PT is enabled.

interface-name—(Optional) Display L2PT information for the specified interface.

Required Privilege Level view

Related Documentation

- show ethernet-switching layer2-protocol-tunneling statistics on page 394
- show ethernet-switching layer2-protocol-tunneling vlan on page 397
- *Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*
- show ethernet-switching layer2-protocol-tunneling statistics on page 394
- show ethernet-switching layer2-protocol-tunneling vlan on page 397
- *Configuring Layer 2 Protocol Tunneling*

List of Sample Output show ethernet-switching layer2-protocol-tunneling interface on page 393
show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0 on page 393

Output Fields Table 35 on page 392 lists the output fields for the **show ethernet-switching layer2-protocol-tunneling interface** command. Output fields are listed in the approximate order in which they appear.

Table 35: show ethernet-switching layer2-protocol-tunneling interface Output Fields

Field Name	Field Description
Interface	Name of an interface on the switch.
Operation	Type of operation being performed on the interface. Values are **Encapsulation** and **Decapsulation**.
State	State of the interface. Values are **active** and **shutdown**.
Description	If the interface state is **shutdown**, displays why the interface is shut down. If the description says **Loop detected**, it means that the interface is an access interface that has received L2PT-enabled PDUs. Access interfaces should not receive L2PT-enabled PDUs. This scenario might mean that there is a loop in the network.

Sample Output

show ethernet-switching layer2-protocol-tunneling interface

```
user@switch> show ethernet-switching layer2-protocol-tunneling interface

Layer2 Protocol Tunneling information:
Interface     Operation       State      Description
xe-0/0/0.0    Encapsulation   Shutdown   Shutdown threshold exceeded
xe-0/0/1.0    Decapsulation   Shutdown   Loop detected
xe-0/0/2.0    Decapsulation   Active
```

show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0

```
user@switch> show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0

Layer2 Protocol Tunneling information:
Interface     Operation       State      Description
xe-0/0/0.0    Encapsulation   Shutdown   Shutdown threshold exceeded
```

show ethernet-switching layer2-protocol-tunneling statistics

Syntax

show ethernet-switching-layer2-protocol-tunneling statistics
<interface *interface-name*>
<vlan *vlan-name*>

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Display Layer 2 protocol tunneling (L2PT) statistics for Layer 2 PDU packets received by the switch.

NOTE: **The show ethernet-switching-layer2-protocol-tunneling statistics command does not display L2PT statistics for Layer 2 PDU packets transmitted from the switch.**

Options

none—Display L2PT statistics for all interfaces on which you enabled L2PT.

interface *interface-name*—(Optional) Display L2PT statistics for the specified interface.

vlan *vlan-name*—(Optional) Display L2PT statistics for the specified VLAN.

Required Privilege Level

view

Related Documentation

- clear ethernet-switching layer2-protocol-tunneling statistics on page 384
- show ethernet-switching layer2-protocol-tunneling interface on page 392
- show ethernet-switching layer2-protocol-tunneling vlan on page 397
- *show vlans*
- *Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches*
- *Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*
- show vlans on page 448
- *Configuring Layer 2 Protocol Tunneling*

List of Sample Output

show ethernet-switching layer2-protocol-tunneling statistics on page 395
show ethernet-switching layer2-protocol-tunneling statistics interface xe-0/0/0.0 on page 395
show ethernet-switching layer2-protocol-tunneling statistics vlan v2 on page 395

Output Fields

Table 36 on page 395 lists the output fields for the **show ethernet-switching layer2-protocol-tunneling statistics** command. Output fields are listed in the approximate order in which they appear.

Table 36: show ethernet-switching layer2-protocol-tunneling statistics Output Fields

VLAN	Field Description
VLAN	Name of a VLAN on which L2PT has been configured.
Interface	Name of an interface on which L2PT has been configured.
Protocol	Name of a protocol for which L2PT has been enabled. Values are **all**, **802.1x**, **802.3ah**, **cdp**, **e-lmi**, **gvrp**, **lacp**, **lldp**, **mmrp**, **mvrp**, **stp**, **udld**, **vstp**, and **vtp**.
Operation	Type of operation being performed on the interface. Values are **Encapsulation** and **Decapsulation**.
Packets	Number of packets that have been encapsulated or de-encapsulated.
Drops	Number of packets that have exceeded the drop threshold and have been dropped.
Shutdowns	Number of times that packets have exceeded the shutdown threshold and the interface has been shut down.

Sample Output

show ethernet-switching layer2-protocol-tunneling statistics

```
user@switch> show ethernet-switching layer2-protocol-tunneling statistics

Layer2 Protocol Tunneling Statistics:
VLAN   Interface  Protocol  Operation     Packets Drops      Shutdowns
v1     xe-0/0/0.0 mvrp      Encapsulation 0       0          0
v1     xe-0/0/1.0 mvrp      Decapsulation 0       0          0
v1     xe-0/0/2.0 mvrp      Decapsulation 60634   0          0
v2     xe-0/0/0.0 cdp       Encapsulation 0       0          0
v2     xe-0/0/0.0 gvrp      Encapsulation 0       0          0
v2     xe-0/0/0.0 lldp      Encapsulation 0       0          0
```

show ethernet-switching layer2-protocol-tunneling statistics interface xe-0/0/0.0

```
user@switch> show ethernet-switching layer2-protocol-tunneling statistics interface xe-0/0/0.0

Layer2 Protocol Tunneling Statistics:
VLAN    Interface  Protocol Operation      Packets  Drops     Shutdowns
v1      xe-0/0/0.0 mvrp     Encapsulation  0        0          0
v2      xe-0/0/0.0 cdp      Encapsulation  0        0          0
v2      xe-0/0/0.0 gvrp     Encapsulation  0        0          0
v2      xe-0/0/0.0 lldp     Encapsulation  0        0          0
v2      xe-0/0/0.0 mvrp     Encapsulation  0        0          0
v2      xe-0/0/0.0 stp      Encapsulation  0        0          0
v2      xe-0/0/0.0 vtp      Encapsulation  0        0          0
v2      xe-0/0/0.0 vstp     Encapsulation  0        0          0
```

show ethernet-switching layer2-protocol-tunneling statistics vlan v2

```
user@switch> show ethernet-switching layer2-protocol-tunneling statistics vlan v2

Layer2 Protocol Tunneling Statistics:
VLAN    Interface   Protocol Operation       Packets  Drops   Shutdowns
v2      xe-0/0/0.0  cdp      Encapsulation   0        0       0
v2      xe-0/0/0.0  gvrp     Encapsulation   0        0       0
```

```
v2      xe-0/0/0.0  lldp     Encapsulation   0        0        0
v2      xe-0/0/0.0  mvrp     Encapsulation   0        0        0
v2      xe-0/0/0.0  stp      Encapsulation   0        0        0
v2      xe-0/0/0.0  vtp      Encapsulation   0        0        0
v2      xe-0/0/0.0  vstp     Encapsulation   0        0        0
v2      xe-0/0/1.0  cdp      Decapsulation   0        0        0
v2      xe-0/0/1.0  gvrp     Decapsulation   0        0        0
v2      xe-0/0/1.0  lldp     Decapsulation   0        0        0
v2      xe-0/0/1.0  mvrp     Decapsulation   0        0        0
v2      xe-0/0/1.0  stp      Decapsulation   0        0        0
v2      xe-0/0/1.0  vtp      Decapsulation   0        0        0
```

show ethernet-switching layer2-protocol-tunneling vlan

Syntax show ethernet-switching-layer2-protocol-tunneling vlan <*vlan-name*>

Release Information Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Display information about Layer 2 protocol tunneling (L2PT) on VLANs that have been configured for L2PT.

Options **none**—Display information about L2PT for the VLANs on which you have configured L2PT.

vlan-name—(Optional) Display information about L2PT for the specified VLAN.

Required Privilege Level view

Related Documentation

- show ethernet-switching layer2-protocol-tunneling interface on page 392
- show ethernet-switching layer2-protocol-tunneling statistics on page 394
- *show vlans*
- *Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches*
- *Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)*
- show vlans on page 448
- *Configuring Layer 2 Protocol Tunneling*

List of Sample Output
show ethernet-switching layer2-protocol-tunneling vlan on page 398
show ethernet-switching layer2-protocol-tunneling vlan v2 on page 398

Output Fields Table 37 on page 397 lists the output fields for the **show ethernet-switching layer2-protocol-tunneling vlan** command. Output fields are listed in the approximate order in which they appear.

Table 37: show ethernet-switching layer2-protocol-tunneling vlan Output Fields

Field Name	Field Description
VLAN	Name of the VLAN on which L2PT has been configured.
Protocol	Name of a protocol for which L2PT has been enabled. Values are **all**, **802.1x**, **802.3ah**, **cdp**, **e-lmi**, **gvrp**, **lacp**, **lldp**, **mmrp**, **mvrp**, **stp**, **vstp**, and **vtp**.
Drop Threshold	Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the VLAN before the switch begins dropping the Layer 2 PDUs.
Shutdown Threshold	Maximum number of Layer 2 PDUs of the specified protocol that can be received per second on the VLAN before the interface is disabled.

Sample Output

show ethernet-switching layer2-protocol-tunneling vlan

```
user@switch> show ethernet-switching layer2-protocol-tunneling vlan

Layer2 Protocol Tunneling VLAN information:
VLAN          Protocol      Drop       Shutdown
                            Threshold  Threshold
v1            mvrp          100        200
v2            cdp           0          0
v2            cdp           0          0
v2            gvrp          0          0
```

show ethernet-switching layer2-protocol-tunneling vlan v2

```
user@switch> show ethernet-switching layer2-protocol-tunneling vlan v2

Layer2 Protocol Tunneling VLAN information:
VLAN          Protocol      Drop       Shutdown
                            Threshold  Threshold
v2            cdp           0          0
v2            cdp           0          0
v2            gvrp          0          0
```

show ethernet-switching mac-learning-log

Syntax show ethernet-switching mac-learning-log

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Displays the event log of learned MAC addresses.

Required Privilege Level view

Related Documentation
- show ethernet-switching table on page 408
- show ethernet-switching interfaces on page 388

List of Sample Output show ethernet-switching mac-learning-log on page 399

Output Fields Table 38 on page 399 lists the output fields for the **show ethernet-switching mac-learning-log** command. Output fields are listed in the approximate order in which they appear.

Table 38: show ethernet-switching mac-learning-log Output Fields

Field Name	Field Description
Date and Time	Timestamp when the MAC address was added or deleted from the log.
Interface name	Interface name: A value defined by the user for configured interfaces. MAC flags are displayed for debugging purposes.
vlan_name	VLAN name. A value defined by the user for all user-configured VLANs.
MAC	Learned MAC address.
Added, learned, deleted, changed, or moved.	MAC address added, learned, deleted, changed or moved from or added to the MAC learning log. For example, a change event is logged when a dynamic entry is changed to a static one. A mac move event is logged and only in this case. A change event is also logged when a MAC moves from one interface to another interface.

Sample Output

show ethernet-switching mac-learning-log

```
user@switch> show ethernet-switching mac-learning-log
Mon Feb 25 08:07:05 2008
 vlan_name v1 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v9 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name HR_vlan mac 00:00:00:00:00:00 was deleted
```

```
Mon Feb 25 08:07:05 2008
 vlan_name v3 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v12 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v13 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name sales_vlan mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name employee1 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name employee2 mac 00:00:00:00:00:00 was deleted
Mon Feb 25 08:07:05 2008
 vlan_name v3 mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name HR_vlan mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name employee2 mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name employee1 mac 00:00:00:00:00:00 was added
Mon Feb 25 08:07:05 2008
 vlan_name employee2 mac 00:00:05:00:00:05 was learned
Mon Feb 25 08:07:05 2008
 vlan_name employee1 mac 00:30:48:90:54:89 was learned
Mon Feb 25 08:07:05 2008
 vlan_name HR_vlan mac 00:00:5e:00:01:00 was learned
Mon Feb 25 08:07:05 2008
 vlan_name sales_vlan mac 00:00:5e:00:01:08 was learned
[output truncated]
```

show ethernet-switching mac-notification

Syntax show ethernet-switching mac-notification

Release Information Command introduced in Junos OS Release 9.6 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display information about MAC notification.

Required Privilege Level view

Related Documentation
- *Verifying That MAC Notification Is Working Properly*

List of Sample Output
show ethernet-switching mac-notification (MAC Notification Enabled) on page 401
show ethernet-switching mac-notification (MAC Notification Disabled) on page 401

Output Fields Table 39 on page 401 lists the output fields for the **show ethernet-switching mac-notification** command. Output fields are listed in the order in which they appear.

Table 39: show ethernet-switching mac-notification Output Fields

Field Name	Field Description
Notification Status	MAC notification status: • **Enabled**—MAC notification is enabled. • **Disabled**—MAC notification is disabled.
Notification Interval	MAC notification interval in seconds.

Sample Output

show ethernet-switching mac-notification (MAC Notification Enabled)

```
user@switch> show ethernet-switching mac-notification
Notification Status                : Enabled
Notification Interval              : 30
```

Sample Output

show ethernet-switching mac-notification (MAC Notification Disabled)

```
user@switch> show ethernet-switching mac-notification
Notification Status                : Disabled
Notification Interval              : 0
```

show ethernet-switching statistics aging

Syntax show ethernet-switching statistics aging
<brief | detail>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display media access control (MAC) aging statistics.

Options **none**—(Optional) Display MAC aging statistics.

brief | detail—(Optional) Display the specified level of output.

Required Privilege Level view

Related Documentation
- show ethernet-switching statistics mac-learning on page 404
- mac-table-aging-time on page 285
- Configuring MAC Table Aging on page 240

List of Sample Output show ethernet-switching statistics aging on page 403

Output Fields Table 40 on page 402 lists the output fields for the **show ethernet-switching statistics aging** command. Output fields are listed in the approximate order in which they appear.

Table 40: show ethernet-switching statistics aging Output Fields

Field Name	Field Description	Level of Output
Total age messages received	Total number of aging messages received from the hardware.	All levels
Immediate aging	Aging message indicating that the entry should be removed immediately.	All levels
MAC address seen	Aging message indicating that the MAC address has been detected by hardware and that the aging timer should be stopped.	All levels
MAC address not seen	Aging message indicating that the MAC address has not been detected by the hardware and that the aging timer should be started.	All levels
Error age messages	The received aging message contains the following errors: • **Invalid VLAN**—The VLAN of the packet does not exist. • **No such entry**—The MAC address and VLAN pair provided by the aging message does not exist. • **Static entry**—An unsuccessful attempt was made to age out a static MAC entry.	All levels

Sample Output

show ethernet-switching statistics aging

```
user@switch> show ethernet-switching statistics aging

Total age messages received: 0
  Immediate aging: 0, MAC address seen: 0, MAC address not seen: 0
Error age messages: 0
  Invalid VLAN: 0, No such entry: 0, Static entry: 0
```

show ethernet-switching statistics mac-learning

Syntax

show ethernet-switching statistics mac-learning
<brief | detail>
<interface *interface-name*>

Release Information

Command introduced in Junos OS Release 9.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display media access control (MAC) learning statistics.

Options

none—(Optional) Display MAC learning statistics for all interfaces.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

interface *interface-name*—(Optional) Display MAC learning statistics for the specified interface.

Required Privilege Level

view

Related Documentation

- *show ethernet-switching statistics aging*
- *show ethernet-switching mac-learning-log*
- *show ethernet-switching table*
- *show ethernet-switching interfaces*
- *Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch*
- *Example: Setting Up Bridging with Multiple VLANs for EX Series Switches*
- show ethernet-switching statistics aging on page 402
- show ethernet-switching mac-learning-log on page 399
- show ethernet-switching table on page 408
- show ethernet-switching interfaces on page 388
- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102

List of Sample Output

Output Fields

Table 41 on page 405 lists the output fields for the **show ethernet-switching statistics mac-learning** command. Output fields are listed in the approximate order in which they appear.

Table 41: show ethernet-switching statistics mac-learning Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface for which statistics are being reported. (Displayed in the output under the heading **Interface**.)	All levels
Learning message from local packets	MAC learning message generated due to packets coming in on the management interface. (Displayed in the output under the heading **Local pkts**.)	All levels
Learning message from transit packets	MAC learning message generated due to packets coming in on network interfaces. (Displayed in the output under the heading **Transit pkts**.)	All levels
Learning message with error	MAC learning messages received with errors (Displayed under the heading **Error**): • **Invalid VLAN**—The VLAN of the packet does not exist. • **Invalid MAC**—The MAC address is either NULL or a multicast MAC address. • **Security violation**—The MAC address is not an allowed MAC address. • **Interface down**—The MAC address is learned on an interface that is down. • **Incorrect membership**—The MAC address is learned on an interface that is not a member of the VLAN. • **Interface limit**—The number of MAC addresses learned on the interface has exceeded the limit. • **MAC move limit**—This MAC address has moved among multiple interfaces too many times in a given interval. • **VLAN limit**—The number of MAC addresses learned on the VLAN has exceeded the limit. • **VLAN membership limit**—The number of MAC addresses learned on the interface as a member of the specified VLAN (VLAN membership MAC limit) has exceeded the limit. • **Invalid VLAN index**—The VLAN of the packet, although configured, does not yet exist in the kernel. • **Interface not learning**—The MAC address is learned on an interface that does not yet allow learning—for example, the interface is blocked. • **No nexthop**—The MAC address is learned on an interface that does not have a unicast next hop. • **MAC learning disabled**—The MAC address is learned on an interface on which MAC learning has been disabled. • **Others**—The message contains some other error.	All levels

Sample Output

show ethernet-switching statistics mac-learning

```
user@switch> show ethernet-switching statistics mac-learning

Learning stats: 0 learn msg rcvd, 0 error
  Interface         Local pkts        Transit pkts      Error
  ge-0/0/0.0        0                 0                 0
  ge-0/0/1.0        0                 0                 0
  ge-0/0/2.0        0                 0                 0
  ge-0/0/3.0        0                 0                 0
```

show ethernet-switching statistics mac-learning detail

```
user@switch> show ethernet-switching statistics mac-learning detail
Learning stats: 0 learn msg rcvd, 0 error

    Interface: ge-0/0/0.0
    Learning message from local packets:   0
    Learning message from transit packets: 1
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                  0
      Security violation:         0       Interface down:               0
      Incorrect membership:       0       Interface limit:              0
      MAC move limit:             0       VLAN limit:                   0
      Invalid VLAN index:         0       Interface not learning:       0
      No nexthop:                 0       MAC learning disabled:        0
      Others:                     0

    Interface: ge-0/0/1.0
    Learning message from local packets:   0
    Learning message from transit packets: 2
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                  0
      Security violation:         0       Interface down:               0
      Incorrect membership:       0       Interface limit:              0
      MAC move limit:             0       VLAN limit:                   0
      Invalid VLAN index:         0       Interface not learning:       0
      No nexthop:                 0       MAC learning disabled:        0
      Others:                     0
```

show ethernet-switching statistics mac-learning interface ge-0/0/28 detail

```
user@switch> show ethernet-switching statistics mac-learning interface ge-0/0/28 detail

    Interface: ge-0/0/28.0
    Learning message from local packets:   0
    Learning message from transit packets: 5
    Learning message with error:           0
      Invalid VLAN:               0       Invalid MAC:                  0
      Security violation:         0       Interface down:               0
      Incorrect membership:       0       Interface limit:              0
      MAC move limit:             0       VLAN limit:                   0
                                          VLAN membership limit:        20
      Invalid VLAN index:         0       Interface not learning:       0
      No nexthop:                 0       MAC learning disabled:        0
      Others:                     0
```

show ethernet-switching statistics mac-learning interface

```
user@switch> show ethernet-switching statistics mac-learning interface ge-0/0/1
Interface         Local pkts        Transit pkts      Error
ge-0/0/1.0        0                 1                 1
```

show ethernet-switching statistics mac-learning detail (QFX Series)

```
user@switch> show ethernet-switching statistics mac-learning detail
Learning stats: 0 learn msg rcvd, 0 error

    Interface: xe-0/0/0.0
    Learning message from local packets:   0
    Learning message from transit packets: 1
    Learning message with error:           0
```

```
  Invalid VLAN:               0       Invalid MAC:                0
  Security violation:         0       Interface down:             0
  Incorrect membership:       0       Interface limit:            0
  MAC move limit:             0       VLAN limit:                 0
  Invalid VLAN index:         0       Interface not learning:     0
  No nexthop:                 0       MAC learning disabled:      0
  Others:                     0

Interface: xe-0/0/1.0
Learning message from local packets:   0
Learning message from transit packets: 2
Learning message with error:           0
  Invalid VLAN:               0       Invalid MAC:                0
  Security violation:         0       Interface down:             0
  Incorrect membership:       0       Interface limit:            0
  MAC move limit:             0       VLAN limit:                 0
  Invalid VLAN index:         0       Interface not learning:     0
  No nexthop:                 0       MAC learning disabled:      0
  Others:                     0
```

show ethernet-switching table

Syntax

show ethernet-switching table
<brief | detail | extensive | summary>
<interface *interface-name*>
<management-vlan>
<sort-by (*name* | *tag*)>
<vlan *vlan-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.
Output for private VLANs introduced in Junos OS Release 12.1 for the QFX Series.

Description

Displays the Ethernet switching table.

Options

none—(Optional) Display brief information about the Ethernet switching table.

brief | detail | extensive | summary—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display the Ethernet switching table for a specific interface.

management-vlan—(Optional) Display the Ethernet switching table for a management VLAN.

sort-by (*name* | *tag*)—(Optional) Display VLANs in ascending order of VLAN IDs or VLAN names.

vlan *vlan-name*—(Optional) Display the Ethernet switching table for a specific VLAN.

Required Privilege Level

view

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102

List of Sample Output

Output Fields

Table 42 on page 408 lists the output fields for the **show ethernet-switching table** command. Output fields are listed in the approximate order in which they appear.

Table 42: show ethernet-switching table Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of a VLAN.	All levels

Table 42: show ethernet-switching table Output Fields *(continued)*

Field Name	Field Description	Level of Output
MAC address	MAC address associated with the VLAN.	All levels
Type	Type of MAC address: • **static**—The MAC address is manually created. • **learn**—The MAC address is learned dynamically from a packet's source MAC address. • **flood**—The MAC address is unknown and flooded to all members.	All levels
Age	Time remaining before the entry ages out and is removed from the Ethernet switching table.	All levels
Interfaces	Interface associated with learned MAC addresses or with the **All-members** option (flood entry).	All levels
Learned	For learned entries, the time at which the entry was added to the Ethernet switching table.	**detail**, **extensive**

Sample Output

show ethernet-switching table

```
user@switch> show ethernet-switching table
Ethernet-switching table: 57 entries, 17 learned
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 xe-0/0/44.0
  F2                00:19:e2:50:7d:e0 Static         - Router
  Linux             *                 Flood          - All-members
  Linux             00:19:e2:50:7d:e0 Static         - Router
  Linux             00:30:48:90:54:89 Learn          0 xe-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Learn          0 xe-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T1                00:19:e2:50:7d:e0 Static         - Router
  T10               *                 Flood          - All-members
  T10               00:00:5e:00:01:09 Static         - Router
  T10               00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T10               00:19:e2:50:7d:e0 Static         - Router
  T111              *                 Flood          - All-members
  T111              00:19:e2:50:63:e0 Learn          0 xe-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 xe-0/0/15.0
  T2                *                 Flood          - All-members
  T2                00:00:5e:00:01:01 Static         - Router
  T2                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T2                00:19:e2:50:7d:e0 Static         - Router
  T3                *                 Flood          - All-members
  T3                00:00:5e:00:01:02 Static         - Router
  T3                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T3                00:19:e2:50:7d:e0 Static         - Router
  T4                *                 Flood          - All-members
```

```
  T4                00:00:5e:00:01:03 Static         - Router
  T4                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
[output truncated]
```

show ethernet-switching table (Private VLANs)

```
user@switch> show ethernet-switching table
Ethernet-switching table: 10 entries, 3 learned
  VLAN              MAC address       Type         Age Interfaces
  pvlan             *                 Flood          - All-members
  pvlan             00:10:94:00:00:02 Replicated       - xe-0/0/28.0
  pvlan             00:10:94:00:00:35 Replicated       - xe-0/0/46.0
  pvlan             00:10:94:00:00:46 Replicated       - xe-0/0/4.0
  c2                *                 Flood          - All-members
  c2                00:10:94:00:00:02 Learn          0 xe-0/0/28.0
  c1                *                 Flood          - All-members
  c1                00:10:94:00:00:46 Learn          0 xe-0/0/4.0
  __pvlan_pvlan_xe-0/0/46.0__ *       Flood          - All-members
  __pvlan_pvlan_xe-0/0/46.0__ 00:10:94:00:00:35 Learn        0 xe-0/0/46.0
```

show ethernet-switching table brief

```
user@switch> show ethernet-switching table brief
Ethernet-switching table: 57 entries, 17 learned
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 xe-0/0/44.0
  F2                00:19:e2:50:7d:e0 Static         - Router
  Linux             *                 Flood          - All-members
  Linux             00:19:e2:50:7d:e0 Static         - Router
  Linux             00:30:48:90:54:89 Learn          0 xe-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Learn          0 xe-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T1                00:19:e2:50:7d:e0 Static         - Router
  T10               *                 Flood          - All-members
  T10               00:00:5e:00:01:09 Static         - Router
  T10               00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T10               00:19:e2:50:7d:e0 Static         - Router
  T111              *                 Flood          - All-members
  T111              00:19:e2:50:63:e0 Learn          0 xe-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 xe-0/0/15.0
  T2                *                 Flood          - All-members
  T2                00:00:5e:00:01:01 Static         - Router
  T2                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T2                00:19:e2:50:7d:e0 Static         - Router
  T3                *                 Flood          - All-members
  T3                00:00:5e:00:01:02 Static         - Router
  T3                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
  T3                00:19:e2:50:7d:e0 Static         - Router
  T4                *                 Flood          - All-members
  T4                00:00:5e:00:01:03 Static         - Router
  T4                00:19:e2:50:63:e0 Learn          0 xe-0/0/46.0
[output truncated]
```

show ethernet-switching table detail

```
user@switch> show ethernet-switching table detail
Ethernet-switching table: 57 entries, 17 learned
  F2, *
```

```
  Interface(s): xe-0/0/44.0
  Type: Flood
  Nexthop index: 0

F2, 00:00:05:00:00:03
  Interface(s): xe-0/0/44.0
  Type: Learn, Age: 0, Learned: 2:03:09
  Nexthop index: 0

F2, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
  Nexthop index: 0

Linux, *
  Interface(s): xe-0/0/47.0
  Type: Flood
  Nexthop index: 0

Linux, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
  Nexthop index: 0

Linux, 00:30:48:90:54:89
  Interface(s): xe-0/0/47.0
  Type: Learn, Age: 0, Learned: 2:03:08
  Nexthop index: 0

T1, *
  Interface(s): xe-0/0/46.0
  Type: Flood
  Nexthop index: 0

T1, 00:00:05:00:00:01
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:07
  Nexthop index: 0

T1, 00:00:5e:00:01:00
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T1, 00:19:e2:50:63:e0
  Interface(s): xe-0/0/46.0
  Type: Learn, Age: 0, Learned: 2:03:07
  Nexthop index: 0

T1, 00:19:e2:50:7d:e0
  Interface(s): Router
  Type: Static
  Nexthop index: 0

T10, *
  Interface(s): xe-0/0/46.0
  Type: Flood
  Nexthop index: 0

T10, 00:00:5e:00:01:09
  Interface(s): Router
```

```
    Type: Static
    Nexthop index: 0

  T10, 00:19:e2:50:63:e0
    Interface(s): xe-0/0/46.0
    Type: Learn, Age: 0, Learned: 2:03:08
    Nexthop index: 0

  T10, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  T111, *
    Interface(s): xe-0/0/15.0
    Type: Flood
    Nexthop index: 0
[output truncated]
```

show ethernet-switching table extensive

```
user@switch> show ethernet-switching table extensive
Ethernet-switching table: 57 entries, 17 learned
  F2, *
    Interface(s): xe-0/0/44.0
    Type: Flood
    Nexthop index: 0

  F2, 00:00:05:00:00:03
    Interface(s): xe-0/0/44.0
    Type: Learn, Age: 0, Learned: 2:03:09
    Nexthop index: 0

  F2, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  Linux, *
    Interface(s): xe-0/0/47.0
    Type: Flood
    Nexthop index: 0

  Linux, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  Linux, 00:30:48:90:54:89
    Interface(s): xe-0/0/47.0
    Type: Learn, Age: 0, Learned: 2:03:08
    Nexthop index: 0

  T1, *
    Interface(s): xe-0/0/46.0
    Type: Flood
    Nexthop index: 0

  T1, 00:00:05:00:00:01
    Interface(s): xe-0/0/46.0
    Type: Learn, Age: 0, Learned: 2:03:07
```

```
    Nexthop index: 0

  T1, 00:00:5e:00:01:00
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  T1, 00:19:e2:50:63:e0
    Interface(s): xe-0/0/46.0
    Type: Learn, Age: 0, Learned: 2:03:07
    Nexthop index: 0

  T1, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  T10, *
    Interface(s): xe-0/0/46.0
    Type: Flood
    Nexthop index: 0

  T10, 00:00:5e:00:01:09
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  T10, 00:19:e2:50:63:e0
    Interface(s): xe-0/0/46.0
    Type: Learn, Age: 0, Learned: 2:03:08
    Nexthop index: 0

  T10, 00:19:e2:50:7d:e0
    Interface(s): Router
    Type: Static
    Nexthop index: 0

  T111, *
    Interface(s): xe-0/0/15.0
    Type: Flood
    Nexthop index: 0
[output truncated]
```

show ethernet-switching table interface

```
user@switch> show ethernet-switching table interface xe-0/0/1
Ethernet-switching table: 1 unicast entries
  VLAN              MAC address       Type         Age Interfaces
  V1                *                 Flood          - All-members
  V1                00:00:05:00:00:05 Learn          0 xe-0/0/1.0
```

show interfaces xe

Syntax

show interfaces *device-name:type-fpc/pic/port*
<brief | detail | extensive | terse>
<descriptions>
<media>
<routing-instance (all | *instance-name*)>
<snmp-index *snmp-index*>
<statistics>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display status information about the specified 10-Gigabit Ethernet interface. This command does not display statistics for routed VLAN interfaces.

Options

device-name:type-fpc/pic/port—(QFabric systems only) The device name is either the serial number or the alias of the QFabric system component, such as a Node device, Interconnect device, or QFabric infrastructure. The name must contain a maximum of 128 characters and not contain any colons.

brief | detail | extensive | terse—(Optional) Display the specified level of output.

descriptions—(Optional) Display interface description strings.

media—(Optional) Display media-specific information about network interfaces.

routing-instance (all | *instance-name*)—(Optional) Display the name of an individual routing instance or display all routing instances.

snmp-index *snmp-index*—(Optional) Display information for the specified SNMP index of the interface.

statistics—(Optional) Display static interface statistics.

Required Privilege Level

view

Related Documentation

- *Monitoring Interface Status and Traffic*
- *Troubleshooting Network Interfaces*
- *Troubleshooting an Aggregated Ethernet Interface*
- *Junos OS Network Interfaces Library for Routing Devices*

List of Sample Output

show interfaces terse on page 430
show interfaces (QFabric System) on page 430

Output Fields Table 43 on page 415 lists the output fields for the **show interfaces xe** command. Output fields are listed in the approximate order in which they appear.

Table 43: show interfaces xe Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	All levels
Enabled	State of the interface.	All levels
Interface index	Index number of the physical interface, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Link-level type	Encapsulation being used on the physical interface.	All levels
MTU	Maximum transmission unit size on the physical interface.	All levels
Speed	Speed at which the interface is running.	All levels
Duplex	Duplex mode of the interface, either **Full-Duplex** or **Half-Duplex**.	All levels
Loopback	Loopback status: **Enabled** or **Disabled**. If loopback is enabled, type of loopback: **Local** or **Remote**.	All levels
Source filtering	Source filtering status: **Enabled** or **Disabled**.	All levels
LAN-PHY mode	10-Gigabit Ethernet interface operating in Local Area Network Physical Layer Device (LAN PHY) mode. LAN PHY allows 10-Gigabit Ethernet wide area links to use existing Ethernet applications.	All levels
Unidirectional	Unidirectional link mode status for 10-Gigabit Ethernet interface: **Enabled** or **Disabled** for parent interface; **Rx-only** or **Tx-only** for child interfaces.	All levels
Flow control	Flow control status: **Enabled** or **Disabled**. NOTE: This field is only displayed if asymmetric flow control is not configured.	All levels

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output
Configured-flow-control	Configured flow control for the interface transmit buffers (**tx-buffers**) and receive buffers (**rx-buffers**): • **tx-buffers—On** if the interface is configured to respond to Ethernet PAUSE messages received from the connected peer. **Off** if the interface is not configured to respond to received PAUSE messages. • **rx-buffers—On** if the interface is configured to generate and send Ethernet PAUSE messages to the connected peer. **Off** if the interface is not configured to generate and send PAUSE messages. NOTE: This field is only displayed if asymmetric flow control is configured.	All levels
Auto-negotiation	Autonegotiation status: **Enabled** or **Disabled**.	All levels
Remote-fault	Remote fault status: • **Online**—Autonegotiation is manually configured as online. • **Offline**—Autonegotiation is manually configured as offline.	All levels
Device flags	Information about the physical device.	All levels
Interface flags	Information about the interface.	All levels
Link flags	Information about the link.	All levels
Wavelength	Configured wavelength, in nanometers (nm).	All levels
Frequency	Frequency associated with the configured wavelength, in terahertz (THz).	All levels
CoS queues	Number of CoS queues configured.	**detail extensive** none
Schedulers	Number of CoS schedulers configured.	**extensive**
Hold-times	Current interface hold-time up and hold-time down, in milliseconds.	**detail extensive**
Current address	Configured MAC address.	**detail extensive** none
Hardware address	Hardware MAC address.	**detail extensive** none
Last flapped	Date, time, and how long ago the interface went from down to up. The format is **Last flapped:** ***year-month-day hour: :minute:second:timezone*** **(*hour:minute:second* ago)**. For example, **Last flapped: 2008–01–16 10:52:40 UTC (3d 22:58 ago).**	**detail extensive** none
Input Rate	Input rate in bits per second (bps) and packets per second (pps).	None specified
Output Rate	Output rate in bps and pps.	None specified
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output
Traffic statistics	Number and rate of bytes and packets received and transmitted on the physical interface. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface. NOTE: The bandwidth bps counter is not enabled.	**detail extensive**
Input errors	Input errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Errors**—Sum of the incoming frame aborts and FCS errors. • **Drops**—Number of packets dropped by the input queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **Framing errors**—Number of packets received with an invalid frame checksum (FCS). • **Runts**—Number of frames received that are smaller than the runt threshold. • **Policed discards**—Number of frames that the incoming packet match code discarded because they were not recognized or not of interest. Usually, this field reports protocols that Junos OS does not handle. • **L3 incompletes**—Number of incoming packets discarded because they failed Layer 3 sanity checks of the header. For example, a frame with less than 20 bytes of available IP header is discarded. L3 incomplete errors can be ignored if you configure the **ignore-l3-incompletes** statement. • **L2 channel errors**—Number of times the software did not find a valid logical interface for an incoming frame. • **L2 mismatch timeouts**—Number of malformed or short packets that caused the incoming packet handler to discard the frame as unreadable. • **FIFO errors**—Number of FIFO errors in the receive direction that are reported by the ASIC on the PIC. If this value is ever nonzero, the PIC is probably malfunctioning. • **Resource errors**—Sum of transmit drops.	**extensive**

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output
Output errors	Output errors on the interface. The following paragraphs explain the counters whose meaning might not be obvious: • **Carrier transitions**—Number of times the interface has gone from **down** to **up**. This number does not normally increment quickly, increasing only when the cable is unplugged, the far-end system is powered down and then up, or another problem occurs. If the number of carrier transitions increments quickly (perhaps once every 10 seconds), the cable, the far-end system, or the PIC or PIM is malfunctioning. • **Errors**—Sum of the outgoing frame aborts and FCS errors. • **Drops**—Number of packets dropped by the output queue of the I/O Manager ASIC. If the interface is saturated, this number increments once for every packet that is dropped by the ASIC's RED mechanism. • **Collisions**—Number of Ethernet collisions. The Gigabit Ethernet PIC supports only full-duplex operation, so for Gigabit Ethernet PICs, this number should always remain 0. If it is nonzero, there is a software bug. • **Aged packets**—Number of packets that remained in shared packet SDRAM so long that the system automatically purged them. The value in this field should never increment. If it does, it is most likely a software bug or possibly malfunctioning hardware. • **FIFO errors**—Number of FIFO errors in the send direction as reported by the ASIC on the PIC. If this value is ever nonzero, the PIC is probably malfunctioning. • **HS link CRC errors**—Number of errors on the high-speed links between the ASICs responsible for handling the router interfaces. • **MTU errors**—Number of packets whose size exceeded the MTU of the interface. • **Resource errors**—Sum of transmit drops.	**extensive**
Egress queues	Total number of egress queues supported on the specified interface.	**detail extensive**
Queue counters (Egress)	CoS queue number and its associated user-configured forwarding class name. • **Queued packets**—Number of queued packets. • **Transmitted packets**—Number of transmitted packets. • **Dropped packets**—Number of packets dropped by the ASIC's RED mechanism.	**detail extensive**
Queue Number	The CoS queue number and the forwarding classes mapped to the queue number. The **Mapped forwarding class** column lists the forwarding classes mapped to each CoS queue.	**detail extensive**
Ingress queues	Total number of ingress queues supported on the specified interface.	**extensive**
Queue counters (Ingress)	CoS queue number and its associated user-configured forwarding class name. • **Queued packets**—Number of queued packets. • **Transmitted packets**—Number of transmitted packets. • **Dropped packets**—Number of packets dropped by the ASIC's RED mechanism.	**extensive**

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output
Active alarms and Active defects	Ethernet-specific defects that can prevent the interface from passing packets. When a defect persists for a certain amount of time, it is promoted to an alarm. Based on the switch configuration, an alarm can ring the red or yellow alarm bell on the switch, or turn on the red or yellow alarm LED on the craft interface. These fields can contain the value **None** or **Link**. • **None**—There are no active defects or alarms. • **Link**—Interface has lost its link state, which usually means that the cable is unplugged, the far-end system has been turned off, or the PIC is malfunctioning.	**detail extensive** none
PCS statistics	Physical Coding Sublayer (PCS) fault conditions from the LAN PHY device.	**detail extensive**
MAC statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC subsystem. • **Total octets** and **total packets**—Total number of octets and packets. For Gigabit Ethernet IQ PICs, the received octets count varies by interface type. • **Unicast packets, Broadcast packets,** and **Multicast packets**—Number of unicast, broadcast, and multicast packets. • **CRC/Align errors**—Total number of packets received that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, and had either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a nonintegral number of octets (Alignment Error). • **FIFO error**—Number of FIFO errors that are reported by the ASIC on the PIC. If this value is ever nonzero, the PIC is probably malfunctioning. • **MAC control frames**—Number of MAC control frames. • **MAC pause frames**—Number of MAC control frames with **pause** operational code. • **Oversized frames**—Number of packets that exceeds the configured MTU. • **Jabber frames**—Number of frames that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error. This definition of jabber is different from the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition in which any packet exceeds 20 ms. The allowed range to detect jabber is from 20 ms to 150 ms. • **Fragment frames**—Total number of packets that were less than 64 octets in length (excluding framing bits, but including FCS octets), and had either an FCS error or an alignment error. Fragment frames normally increment because both runts (which are normal occurrences caused by collisions) and noise hits are counted. • **VLAN tagged frames**—Number of frames that are VLAN tagged. The system uses the TPID of 0x8100 in the frame to determine whether a frame is tagged or not. This counter is not supported on EX Series switches and is always displayed as 0. • **Code violations**—Number of times an event caused the PHY to indicate “Data reception error” or “invalid data symbol error.”	**extensive**
Filter statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC address filter subsystem.	**extensive**

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output

Autonegotiation information	Information about link autonegotiation.
	• **Negotiation status:**
	  • **Incomplete**—Ethernet interface has the speed or link mode configured.
	  • **No autonegotiation**—Remote Ethernet interface has the speed or link mode configured, or does not perform autonegotiation.
	  • **Complete**—Ethernet interface is connected to a device that performs autonegotiation and the autonegotiation process is successful.
	• **Link partner status**—**OK** when the Ethernet interface is connected to a device that performs autonegotiation and the autonegotiation process is successful.
	• **Link partner:**
	  • **Link mode**—Depending on the capability of the attached Ethernet device, either **Full-duplex** or **Half-duplex**.
	  • **Flow control**—Types of flow control supported by the remote Ethernet device. For Fast Ethernet interfaces, the type is **None**. For Gigabit Ethernet interfaces, types are **Symmetric** (link partner supports **PAUSE** on receive and transmit), **Asymmetric** (link partner supports **PAUSE** on transmit), and **Symmetric/Asymmetric** (link partner supports both **PAUSE** on receive and transmit or only **PAUSE** receive).
	  • **Remote fault**—Remote fault information from the link partner—**Failure** indicates a receive link error. **OK** indicates that the link partner is receiving. **Negotiation error** indicates a negotiation error. **Offline** indicates that the link partner is going offline.
	• **Local resolution**:
	  • **Flow control**—Types of flow control supported by the remote Ethernet device. For Gigabit Ethernet interfaces, types are **Symmetric** (link partner supports **PAUSE** on receive and transmit), **Asymmetric** (link partner supports **PAUSE** on transmit), and **Symmetric/Asymmetric** (link partner supports both **PAUSE** on receive and transmit or only **PAUSE** receive).
	For asymmetric **PAUSE**, shows if the PAUSE transmit and **PAUSE** receive states on the interface are **enable** or **disable**.
	  • **Remote fault**—Remote fault information. **Link OK** (no error detected on receive), **Offline** (local interface is offline), and **Link Failure** (link error detected on receive).
		extensive

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output
Packet Forwarding Engine configuration	Information about the configuration of the Packet Forwarding Engine: • **Destination slot**—FPC slot number. • **CoS transmit queue**—Queue number and its associated user-configured forwarding class name. • **Bandwidth %**—Percentage of bandwidth allocated to the queue. • **Bandwidth bps**—Bandwidth allocated to the queue (in bps). • **Buffer %**—Percentage of buffer space allocated to the queue. • **Buffer usec**—Amount of buffer space allocated to the queue, in microseconds. This value is nonzero only if the buffer size is configured in terms of time. • **Priority**—Queue priority: **low** or **high**. • **Limit**—Displayed if rate limiting is configured for the queue. Possible values are **none** and **exact**. If **exact** is configured, the queue transmits only up to the configured bandwidth, even if excess bandwidth is available. If **none** is configured, the queue transmits beyond the configured bandwidth if bandwidth is available.	**extensive**
Logical Interface		
Logical interface	Name of the logical interface.	All levels
Index	Index number of the logical interface, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP interface index number for the logical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Flags	Information about the logical interface.	All levels
Encapsulation	Encapsulation on the logical interface.	All levels
Protocol	Protocol family.	**detail extensive** none
Traffic statistics	Number and rate of bytes and packets received (input) and transmitted (output) on the specified interface.	**detail extensive**
IPv6 transit statistics	If IPv6 statics tracking is enabled, number of IPv6 bytes and packets received and transmitted on the logical interface.	**extensive**
Local statistics	Number and rate of bytes and packets destined to and from the switch.	**extensive**
Transit statistics	Number and rate of bytes and packets transiting the switch.	**extensive**
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Route Table	Route table in which the logical interface address is located. For example, **0** refers to the routing table inet.0.	**detail extensive** none

Table 43: show interfaces xe Output Fields *(continued)*

Field Name	Field Description	Level of Output
Input Filters	Names of any input filters applied to this interface.	**detail extensive**
Output Filters	Names of any output filters applied to this interface.	**detail extensive**
Flags	Information about protocol family flags. If unicast Reverse Path Forwarding (uRPF) is explicitly configured on the specified interface, the uRPF flag appears. If uRPF was configured on a different interface (and therefore is enabled on all switch interfaces) but was not explicitly configured on the specified interface, the uRPF flag does not appear even though uRPF is enabled.	**detail extensive**
Addresses, Flags	Information about the address flags.	**detail extensive** none
protocol-family	Protocol family configured on the logical interface. If the protocol is **inet**, the IP address of the interface is also displayed.	**brief**
Flags	Information about the address flag.	**detail extensive** none
Destination	IP address of the remote side of the connection.	**detail extensive** none
Local	IP address of the logical interface.	**detail extensive** none
Broadcast	Broadcast address of the logical interlace.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**

Sample Output

show interfaces

```
user@switch> show interfaces xe-0/0/1
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Flow control: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
  Last flapped   : 2011-06-01 00:42:03 PDT (00:02:42 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

  Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Input packets : 0
```

```
    Output packets: 0
    Protocol eth-switch, MTU: 0
      Flags: Trunk-Mode
```

show interfaces (Asymmetric Flow Control)

```
user@switch> show interfaces xe-0/0/1
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Configured-flow-control  tx-buffers: off  rx-buffers: on
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
  Last flapped   : 2011-06-01 00:42:03 PDT (00:02:42 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

  Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Protocol eth-switch, MTU: 0
      Flags: Trunk-Mode
```

show interfaces brief

```
user@switch> show interfaces xe-0/0/1 brief
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None

  Logical interface xe-0/0/1.0
    Flags: SNMP-Traps Encapsulation: ENET2
    eth-switch
```

show interfaces detail

```
user@switch> show interfaces xe-0/0/1 detail
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591, Generation: 169
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Flow control: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
```

```
Last flapped   : 2011-06-01 00:42:03 PDT (00:02:50 ago)
Statistics last cleared: 2011-06-01 00:44:39 PDT (00:00:14 ago)
Traffic statistics:
 Input  bytes  :                    0                    0 bps
 Output bytes  :                    0                    0 bps
 Input  packets:                    0                    0 pps
 Output packets:                    0                    0 pps
 IPv6 transit statistics:
  Input  bytes  :                   0
  Output bytes  :                   0
  Input  packets:                   0
  Output packets:                   0
Egress queues: 12 supported, 9 in use
Queue counters:       Queued packets  Transmitted packets      Dropped packets

  0 best-effort                    0                    0                    0

  1 fc7                            0                    0                    0

  2 no-loss                        0                    0                    0

  3 fcoe                           0                    0                    0

  4 fc4                            0                    0                    0

  5 fc5                            0                    0                    0

  6 fc6                            0                    0                    0

  7 network-cont                   0                    0                    0

  8 mcast                          0                    0                    0

Queue number:         Mapped forwarding classes
  0                   best-effort
  1                   fc7
  2                   no-loss
  3                   fcoe
  4                   fc4
  5                   fc5
  6                   fc6
  7                   network-control
  8                   mcast
Active alarms  : None
Active defects : None

Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523) (Generation 143)
  Flags: SNMP-Traps 0x0 Encapsulation: ENET2
  Traffic statistics:
   Input  bytes  :                  0
   Output bytes  :                  0
   Input  packets:                  0
   Output packets:                  0
  Local statistics:
   Input  bytes  :                  0
   Output bytes  :                  0
   Input  packets:                  0
   Output packets:                  0
  Transit statistics:
   Input  bytes  :                  0                    0 bps
   Output bytes  :                  0                    0 bps
```

```
    Input  packets:                    0                    0 pps
    Output packets:                    0                    0 pps
   Protocol eth-switch, MTU: 0, Generation: 170, Route table: 0
     Flags: Trunk-Mode
```

show interfaces detail (Asymmetric Flow Control)

```
user@switch> show interfaces xe-0/0/1 detail
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591, Generation: 169
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Configured-flow-control  tx-buffers: off  rx-buffers: on
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
  Last flapped   : 2011-06-01 00:42:03 PDT (00:02:50 ago)
  Statistics last cleared: 2011-06-01 00:44:39 PDT (00:00:14 ago)
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :                   0
    Input  packets:                   0
    Output packets:                   0
  Egress queues: 12 supported, 9 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                   0                    0                    0

    1 fc7                           0                    0                    0

    2 no-loss                       0                    0                    0

    3 fcoe                          0                    0                    0

    4 fc4                           0                    0                    0

    5 fc5                           0                    0                    0

    6 fc6                           0                    0                    0

    7 network-cont                  0                    0                    0

    8 mcast                         0                    0                    0

  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   fc7
    2                   no-loss
    3                   fcoe
    4                   fc4
    5                   fc5
    6                   fc6
```

```
    7                   network-control
    8                   mcast
  Active alarms  : None
  Active defects : None

  Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523) (Generation 143)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                    0
     Output bytes  :                    0
     Input  packets:                    0
     Output packets:                    0
    Local statistics:
     Input  bytes  :                    0
     Output bytes  :                    0
     Input  packets:                    0
     Output packets:                    0
    Transit statistics:
     Input  bytes  :                    0                    0 bps
     Output bytes  :                    0                    0 bps
     Input  packets:                    0                    0 pps
     Output packets:                    0                    0 pps
    Protocol eth-switch, MTU: 0, Generation: 170, Route table: 0
      Flags: Trunk-Mode
```

show interfaces extensive

```
user@switch> show interfaces xe-0/0/1 extensive
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591, Generation: 169
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Flow control: Disabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
  Last flapped   : 2011-06-01 00:42:03 PDT (00:03:08 ago)
  Statistics last cleared: 2011-06-01 00:44:39 PDT (00:00:32 ago)
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                    0
    Output bytes  :                    0
    Input  packets:                    0
    Output packets:                    0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3
incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0,
Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
 FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Egress queues: 12 supported, 9 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets
```

```
  0 best-effort                 0                 0                 0

  1 fc7                         0                 0                 0

  2 no-loss                     0                 0                 0

  3 fcoe                        0                 0                 0

  4 fc4                         0                 0                 0

  5 fc5                         0                 0                 0

  6 fc6                         0                 0                 0

  7 network-cont                0                 0                 0

  8 mcast                       0                 0                 0

Queue number:         Mapped forwarding classes
  0                   best-effort
  1                   fc7
  2                   no-loss
  3                   fcoe
  4                   fc4
  5                   fc5
  6                   fc6
  7                   network-control
  8                   mcast
Active alarms  : None
Active defects : None
MAC statistics:                      Receive         Transmit
  Total octets                             0                0
  Total packets                            0                0
  Unicast packets                          0                0
  Broadcast packets                        0                0
  Multicast packets                        0                0
  CRC/Align errors                         0                0
  FIFO errors                              0                0
  MAC control frames                       0                0
  MAC pause frames                         0                0
  Oversized frames                         0
  Jabber frames                            0
  Fragment frames                          0
  VLAN tagged frames                       0
  Code violations                          0
MAC Priority Flow Control Statistics:
  Prioity :  0                             0                0
  Prioity :  1                             0                0
  Prioity :  2                             0                0
  Prioity :  3                             0                0
  Prioity :  4                             0                0
  Prioity :  5                             0                0
  Prioity :  6                             0                0
  Prioity :  7                             0                0
Filter statistics:
  Input packet count                       0
  Input packet rejects                     0
  Input DA rejects                         0
  Input SA rejects                         0
  Output packet count                                       0
```

```
    Output packet pad count                                  0
    Output packet error count                                0
    CAM destination filters: 1, CAM source filters: 0
  Packet Forwarding Engine configuration:
    Destination slot: 0
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority
Limit
                              %           bps     %           usec
    0 best-effort            75     7500000000    75              0      low
none
    7 network-control         5      500000000     5              0      low
none
    8 mcast                  20     2000000000    20              0      low
none

  Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523) (Generation 143)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
    Transit statistics:
     Input  bytes  :                   0                  0 bps
     Output bytes  :                   0                  0 bps
     Input  packets:                   0                  0 pps
     Output packets:                   0                  0 pps
    Protocol eth-switch, MTU: 0, Generation: 170, Route table: 0
      Flags: Trunk-Mode
```

show interfaces extensive (Asymmetric Flow Control)

```
user@switch> show interfaces xe-0/0/1 extensive
Physical interface: xe-0/0/1, Enabled, Physical link is Up
  Interface index: 49195, SNMP ifIndex: 591, Generation: 169
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
Disabled,
  Configured-flow-control  tx-buffers: off  rx-buffers: on
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 12 supported, 12 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:1d:b5:f7:4e:e1, Hardware address: 00:1d:b5:f7:4e:e1
  Last flapped   : 2011-06-01 00:42:03 PDT (00:03:08 ago)
  Statistics last cleared: 2011-06-01 00:44:39 PDT (00:00:32 ago)
  Traffic statistics:
   Input  bytes  :                   0                  0 bps
   Output bytes  :                   0                  0 bps
   Input  packets:                   0                  0 pps
   Output packets:                   0                  0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
```

```
    Output bytes  :                   0
    Input  packets:                   0
    Output packets:                   0
  Input errors:
    Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3
incompletes: 0, L2 channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0,
Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
 FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
  Egress queues: 12 supported, 9 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                   0                   0                   0

    1 fc7                           0                   0                   0

    2 no-loss                       0                   0                   0

    3 fcoe                          0                   0                   0

    4 fc4                           0                   0                   0

    5 fc5                           0                   0                   0

    6 fc6                           0                   0                   0

    7 network-cont                  0                   0                   0

    8 mcast                         0                   0                   0

  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   fc7
    2                   no-loss
    3                   fcoe
    4                   fc4
    5                   fc5
    6                   fc6
    7                   network-control
    8                   mcast
  Active alarms  : None
  Active defects : None
  MAC statistics:                      Receive         Transmit
    Total octets                             0                0
    Total packets                            0                0
    Unicast packets                          0                0
    Broadcast packets                        0                0
    Multicast packets                        0                0
    CRC/Align errors                         0                0
    FIFO errors                              0                0
    MAC control frames                       0                0
    MAC pause frames                         0                0
    Oversized frames                         0
    Jabber frames                            0
    Fragment frames                          0
    VLAN tagged frames                       0
    Code violations                          0
  MAC Priority Flow Control Statistics:
    Prioity :  0                             0                0
    Prioity :  1                             0                0
```

```
    Prioity :  2                              0               0
    Prioity :  3                              0               0
    Prioity :  4                              0               0
    Prioity :  5                              0               0
    Prioity :  6                              0               0
    Prioity :  7                              0               0
  Filter statistics:
    Input packet count                        0
    Input packet rejects                      0
    Input DA rejects                          0
    Input SA rejects                          0
    Output packet count                                       0
    Output packet pad count                                   0
    Output packet error count                                 0
    CAM destination filters: 1, CAM source filters: 0
  Packet Forwarding Engine configuration:
    Destination slot: 0
  CoS information:
    Direction : Output
    CoS transmit queue              Bandwidth           Buffer Priority   Limit
                                %           bps     %       usec
    0 best-effort           75     7500000000    75          0      low    none
    7 network-control        5      500000000     5          0      low    none
    8 mcast                 20     2000000000    20          0      low    none

  Logical interface xe-0/0/1.0 (Index 73) (SNMP ifIndex 523) (Generation 143)
    Flags: SNMP-Traps 0x0 Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :                  0
     Output bytes  :                  0
     Input  packets:                  0
     Output packets:                  0
    Local statistics:
     Input  bytes  :                  0
     Output bytes  :                  0
     Input  packets:                  0
     Output packets:                  0
    Transit statistics:
     Input  bytes  :                  0                   0 bps
     Output bytes  :                  0                   0 bps
     Input  packets:                  0                   0 pps
     Output packets:                  0                   0 pps
    Protocol eth-switch, MTU: 0, Generation: 170, Route table: 0
      Flags: Trunk-Mode
```

show interfaces terse

```
user@switch> show interfaces xe-0/0/1 terse
Interface               Admin Link Proto    Local                 Remote

xe-0/0/1                up    up
xe-0/0/1.0              up    up   eth-switch
```

show interfaces (QFabric System)

```
user@switch> show interfaces node1:xe-0/0/0
Physical interface: node1:xe-0/0/0, Enabled, Physical link is Down
  Interface index: 129, SNMP ifIndex: 2884086
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex, BPDU
 Error: None, MAC-REWRITE Error: None,
  Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled
```

```
Interface flags: Internal: 0x4000
CoS queues     : 8 supported, 8 maximum usable queues
Current address: 02:00:09:03:00:00, Hardware address: 02:00:09:03:00:00
Last flapped   : Never
Input rate     : 0 bps (0 pps)
Output rate    : 0 bps (0 pps)
```

show spanning-tree bridge

List of Syntax
Syntax on page 432
Syntax (QFX Series) on page 432

Syntax
show spanning-tree bridge
<brief | detail>
<msti *msti-id*>
<routing-instance *routing-instance-name*>
<vlan-id *vlan-id*>

Syntax (QFX Series)
show spanning-tree bridge
<brief | detail>
<msti *msti-id*>
<vlan-id *vlan-id*>

Release Information
Command introduced in Junos OS Release 8.4.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description
Display the configured or calculated Spanning Tree Protocol (STP) parameters.

Options
none—(Optional) Display brief STP bridge information for all multiple spanning-tree instances (MSTIs).

brief | detail—(Optional) Display the specified level of output.

msti *msti-id*—(Optional) Display STP bridge information for the specified MSTI.

routing-instance *routing-instance-name*—(Optional) Display STP bridge information for the specified routing instance.

vlan-id *vlan-id*—(Optional) Display STP bridge information for the specified VLAN.

Required Privilege Level
view

List of Sample Output
show spanning-tree bridge routing-instance on page 433
show spanning-tree bridge msti on page 434
show spanning-tree bridge vlan-id (MSTP) on page 435
show spanning-tree bridge (RSTP) on page 435
show spanning-tree bridge vlan-id (RSTP) on page 436

Output Fields
Table 44 on page 432 lists the output fields for the **show spanning-tree bridge** command. Output fields are listed in the approximate order in which they appear.

Table 44: show spanning-tree bridge Output Fields

Field Name	Field Description
Routing instance name	Name of the routing instance under which the bridge is configured.
Enabled protocol	Spanning Tree Protocol type enabled.

Table 44: show spanning-tree bridge Output Fields *(continued)*

Field Name	Field Description
Root ID	Bridge ID of the elected spanning-tree root bridge. The bridge ID consists of a configurable bridge priority and the MAC address of the bridge.
Root cost	Calculated cost to reach the root bridge from the bridge where the command is entered.
Root port	Interface that is the current elected root port for this bridge.
CIST regional root	Bridge ID of the elected MSTP regional root bridge.
CIST internal root cost	Calculated cost to reach the regional root bridge from the bridge where the command is entered.
Hello time	Configured number of seconds between transmissions of configuration bridge protocol data units (BPDUs).
Maximum age	Configured maximum expected arrival time of hello bridge protocol data units (BPDUs).
Forward delay	How long an STP bridge port remains in the listening and learning states before transitioning to the forwarding state.
Hop count	Configured maximum number of hops a BPDU can be forwarded in the MSTP region.
Message age	Number of elapsed seconds since the most recent BPDU was received.
Number of topology changes	Total number of STP topology changes detected since the routing device last booted.
Time since last topology change	Number of elapsed seconds since the most recent topology change.
Bridge ID (Local)	Locally configured bridge ID. The bridge ID consists of a configurable bridge priority and the MAC address of the bridge.
Extended system ID	System identifier.
MSTI regional root	Bridge ID of the elected MSTP regional root bridge.

Sample Output

show spanning-tree bridge routing-instance

```
user@host> show spanning-tree bridge routing-instance vs1 detail
STP bridge parameters
Routing instance name              : vs1
Enabled protocol                   : MSTP
```

```
STP bridge parameters for CIST
  Root ID                           : 32768.00:13:c3:9e:c8:80
  Root cost                         : 0
  Root port                         : ge-10/2/0
  CIST regional root                : 32768.00:13:c3:9e:c8:80
  CIST internal root cost           : 22000
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Message age                       : 0
  Number of topology changes        : 1
  Time since last topology change   : 1191 seconds
  Local parameters
    Bridge ID                       : 32768.00:90:69:0b:7f:d1
    Extended system ID              : 1

STP bridge parameters for MSTI 1
  MSTI regional root                : 32769.00:13:c3:9e:c8:80
  Root cost                         : 22000
  Root port                         : ge-10/2/0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Number of topology changes        : 1
  Time since last topology change   : 1191 seconds
  Local parameters
    Bridge ID                       : 32769.00:90:69:0b:7f:d1
    Extended system ID              : 1

STP bridge parameters for MSTI 2
  MSTI regional root                : 32770.00:13:c3:9e:c8:80
  Root cost                         : 22000
  Root port                         : ge-10/2/0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Number of topology changes        : 1
  Time since last topology change   : 1191 seconds
  Local parameters
    Bridge ID                       : 32770.00:90:69:0b:7f:d1
    Extended system ID              : 1
```

show spanning-tree bridge msti

```
user@host> show spanning-tree bridge msti 1 routing-instance vs1 detail
STP bridge parameters
Routing instance name               : vs1
Enabled protocol                    : MSTP

STP bridge parameters for MSTI 1
  MSTI regional root                : 32769.00:13:c3:9e:c8:80
  Root cost                         : 22000
  Root port                         : xe-10/2/0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
```

```
  Number of topology changes        : 1
  Time since last topology change   : 1191 seconds
  Local parameters
    Bridge ID                       : 32769.00:90:69:0b:7f:d1
    Extended system ID              : 1
```

show spanning-tree bridge vlan-id (MSTP)

```
user@host> show spanning-tree bridge vlan-id 1101 routing-instance vs1 detail
STP bridge parameters
Routing instance name               : vs1
Enabled protocol                    : MSTP

STP bridge parameters for CIST
  Root ID                           : 32768.00:13:c3:9e:c8:80
  Root cost                         : 0
  Root port                         : xe-10/2/0
  CIST regional root                : 32768.00:13:c3:9e:c8:80
  CIST internal root cost           : 22000
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Hop count                         : 18
  Message age                       : 0
  Number of topology changes        : 0
  Local parameters
    Bridge ID                       : 32768.00:90:69:0b:7f:d1
    Extended system ID              : 1
    Hello time                      : 2 seconds
    Maximum age                     : 20 seconds
    Forward delay                   : 15 seconds
    Path cost method                : 32 bit
    Maximum hop count               : 20
```

show spanning-tree bridge (RSTP)

```
user@host> show spanning-tree bridge
STP bridge parameters
Routing instance name               : GLOBAL
Enabled protocol                    : RSTP
  Root ID                           : 28672.00:90:69:0b:3f:d0
  Hello time                        : 2 seconds
  Maximum age                       : 20 seconds
  Forward delay                     : 15 seconds
  Message age                       : 0
  Number of topology changes        : 58
  Time since last topology change   : 14127 seconds
  Local parameters
    Bridge ID                       : 28672.00:90:69:0b:3f:d0
    Extended system ID              : 0

 STP bridge parameters for bridge VLAN 10
   Root ID                          : 28672.00:90:69:0b:3f:d0
   Hello time                       : 2 seconds
   Maximum age                      : 20 seconds
   Forward delay                    : 15 seconds
   Message age                      : 0
   Number of topology changes       : 58
   Time since last topology change  : 14127 seconds
   Local parameters
     Bridge ID                      : 28672.00:90:69:0b:3f:d0
```

```
    Extended system ID            : 0

 STP bridge parameters for bridge VLAN 20
   Root ID                         : 28672.00:90:69:0b:3f:d0
   Hello time                      : 2 seconds
   Maximum age                     : 20 seconds
   Forward delay                   : 15 seconds
   Message age                     : 0
   Number of topology changes      : 58
   Time since last topology change : 14127 seconds
   Local parameters
     Bridge ID                     : 28672.00:90:69:0b:3f:d0
     Extended system ID            : 0
```

show spanning-tree bridge vlan-id (RSTP)

```
user@host> show spanning-tree bridge vlan-id 10
STP bridge parameters
Routing instance name               : GLOBAL
Enabled protocol                    : RSTP

 STP bridge parameters for VLAN 10
   Root ID                         : 28672.00:90:69:0b:3f:d0
   Hello time                      : 2 seconds
   Maximum age                     : 20 seconds
   Forward delay                   : 15 seconds
   Message age                     : 0
   Number of topology changes      : 58
   Time since last topology change : 14127 seconds
   Local parameters
     Bridge ID                     : 28672.00:90:69:0b:3f:d0
     Extended system ID            : 0
```

show spanning-tree interface

List of Syntax
Syntax on page 437
Syntax (EX Series Switches and the QFX Series) on page 437

Syntax
show spanning-tree interface
<brief | detail>
<msti *msti-id*>
<routing-instance *routing-instance-name*>
<vlan-id *vlan-id*>

Syntax (EX Series Switches and the QFX Series)
show spanning-tree interface
<brief | detail>
<msti *msti-id*>
<vlan-id *vlan-id*>

Release Information
Command introduced in Junos OS Release 8.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description
Display the configured or calculated interface-level STP parameters.

Options
none—Display brief STP interface information.

brief | detail—(Optional) Display the specified level of output.

msti *msti-id*—(Optional) Display STP interface information for the specified MST instance.

routing-instance *routing-instance-name*—(Optional) Display STP interface information for the specified routing instance.

vlan-id *vlan-id*—(Optional) Display STP interface information for the specified VLAN.

Required Privilege Level
view

List of Sample Output
show spanning-tree interface on page 438
show spanning-tree interface (QFX Series) on page 439
show spanning-tree interface detail on page 439
show spanning-tree interface msti on page 441
show spanning-tree interface vlan-id on page 441
show spanning-tree interface (VSTP) on page 442
show spanning-tree interface vlan-id (VSTP) on page 442

Output Fields
Table 45 on page 437 lists the output fields for the **show spanning-tree Interface** command. Output fields are listed in the approximate order in which they appear.

Table 45: show spanning-tree Interface Output Fields

Field Name	Field Description
Interface name	Interface configured to participate in the STP, RSTP, VSTP, or MSTP instance.

Table 45: show spanning-tree Interface Output Fields *(continued)*

Field Name	Field Description
Port ID	Logical interface identifier configured to participate in the MSTP or VSTP instance.
Designated port ID	Port ID of the designated port for the LAN segment to which this interface is attached.
Designated bridge ID	Bridge ID of the designated bridge for the LAN segment to which this interface is attached.
Port Cost	Configured cost for the interface.
Port State	STP port state: forwarding (**FWD**), blocking (**BLK**), listening, learning, or disabled.
Port Role	MSTP, VSTP, or RSTP port role: designated (**DESG**), backup (**BKUP**), alternate (**ALT**), (**ROOT**), or Root Prevented (**Root-Prev**).
Link type	MSTP, VSTP, or RSTP link type. Shared or point-to-point (pt-pt) and edge or nonedge.
Alternate	Identifies the interface as an MSTP, VSTP, or RSTP alternate root port (**Yes**) or nonalternate root port (**No**).
Boundary Port	Identifies the interface as an MSTP regional boundary port (**Yes**) or nonboundary port (**No**).

Sample Output

show spanning-tree interface

```
user@host> show spanning-tree interface routing-instance vs1 detail
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID       bridge ID          Cost
ae1             128:1        128:1  32768.0090690b47d1      1000  FWD    DESG
ge-2/1/2        128:2        128:2  32768.0090690b47d1     20000  FWD    DESG
ge-2/1/5        128:3        128:3  32768.0090690b47d1     29999  FWD    DESG
ge-2/2/1        128:4       128:26  32768.0013c39ec880     20000  FWD    ROOT
xe-9/2/0        128:5        128:5  32768.0090690b47d1      2000  FWD    DESG
xe-9/3/0        128:6        128:6  32768.0090690b47d1      2000  FWD    DESG

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID       bridge ID          Cost
ae1             128:1        128:1  32769.0090690b47d1      1000  FWD    DESG
ge-2/1/2        128:2        128:2  32769.0090690b47d1     20000  FWD    DESG
ge-2/1/5        128:3        128:3  32769.0090690b47d1     29999  FWD    DESG
ge-2/2/1        128:4       128:26  32769.0013c39ec880     20000  FWD    ROOT
xe-9/2/0        128:5        128:5  32769.0090690b47d1      2000  FWD    DESG
xe-9/3/0        128:6        128:6  32769.0090690b47d1      2000  FWD    DESG
```

```
Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ae1              128:1        128:1  32770.0090690b47d1      1000  FWD    DESG
ge-2/1/2         128:2        128:2  32770.0090690b47d1     20000  FWD    DESG
ge-2/1/5         128:3        128:3  32770.0090690b47d1     29999  FWD    DESG
ge-2/2/1         128:4       128:26  32770.0013c39ec880     20000  FWD    ROOT
xe-9/2/0         128:5        128:5  32770.0090690b47d1      2000  FWD    DESG
xe-9/3/0         128:6        128:6  32770.0090690b47d1      2000  FWD    DESG
```

show spanning-tree interface (QFX Series)

```
user@lf0> show spanning-tree interface routing-instance vs1 detail
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ae1              128:1        128:1  32768.0090690b47d1      1000  FWD    DESG
ge-2/1/2         128:2        128:2  32768.0090690b47d1     20000  FWD    DESG
ge-2/1/5         128:3        128:3  32768.0090690b47d1     29999  FWD    DESG
ge-2/2/1         128:4       128:26  32768.0013c39ec880     20000  FWD    ROOT
xe-9/2/0         128:5        128:5  32768.0090690b47d1      2000  FWD    DESG
xe-9/3/0         128:6        128:6  32768.0090690b47d1      2000  FWD    DESG

Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ae1              128:1        128:1  32769.0090690b47d1      1000  FWD    DESG
ge-2/1/2         128:2        128:2  32769.0090690b47d1     20000  FWD    DESG
ge-2/1/5         128:3        128:3  32769.0090690b47d1     29999  FWD    DESG
ge-2/2/1         128:4       128:26  32769.0013c39ec880     20000  FWD    ROOT
xe-9/2/0    128:5        128:5  32769.0090690b47d1      2000  FWD    DESG
xe-9/3/0    128:6        128:6  32769.0090690b47d1      2000  FWD    DESG

Spanning tree interface parameters for instance 2

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ae1              128:1        128:1  32770.0090690b47d1      1000  FWD    DESG
ge-2/1/2         128:2        128:2  32770.0090690b47d1     20000  FWD    DESG
ge-2/1/5         128:3        128:3  32770.0090690b47d1     29999  FWD    DESG
ge-2/2/1         128:4       128:26  32770.0013c39ec880     20000  FWD    ROOT
xe-9/2/0    128:5        128:5  32770.0090690b47d1      2000  FWD    DESG
xe-9/3/0    128:6        128:6  32770.0090690b47d1      2000  FWD    DESG
```

show spanning-tree interface detail

```
user@host> show spanning-tree interface routing-instance vs1 detail
Spanning tree interface parameters for instance 0

Interface name                : ae1
Port identifier               : 128.1
Designated port ID            : 128.1
Port cost                     : 1000
Port state                    : Forwarding
Designated bridge ID          : 32768.00:90:69:0b:47:d1
Port role                     : Designated
Link type                     : Pt-Pt/NONEDGE
```

```
Boundary port                   : No

Interface name                 : ge-2/1/2
Port identifier                : 128.2
Designated port ID             : 128.2
Port cost                      : 20000
Port state                     : Forwarding
Designated bridge ID           : 32768.00:90:69:0b:47:d1
Port role                      : Designated
Link type                      : Pt-Pt/NONEDGE
Boundary port                   : No

Interface name                 : ge-2/1/5
Port identifier                : 128.3
Designated port ID             : 128.3
Port cost                      : 29999
Port state                     : Forwarding
Designated bridge ID           : 32768.00:90:69:0b:47:d1
Port role                      : Designated
Link type                      : Pt-Pt/NONEDGE
Boundary port                   : No

Interface name                 : ge-2/2/1
Port identifier                : 128.4
Designated port ID             : 128.26
Port cost                      : 20000
Port state                     : Forwarding
Designated bridge ID           : 32768.00:13:c3:9e:c8:80
Port role                      : Root
Link type                      : Pt-Pt/NONEDGE
Boundary port                   : No

Interface name                 : xe-9/2/0
Port identifier                : 128.5
Designated port ID             : 128.5
Port cost                      : 2000
Port state                     : Forwarding
Designated bridge ID           : 32768.00:90:69:0b:47:d1
Port role                      : Designated
Link type                      : Pt-Pt/NONEDGE
Boundary port                   : No

Interface name                 : xe-9/3/0
Port identifier                : 128.6
Designated port ID             : 128.6
Port cost                      : 2000
Port state                     : Forwarding
Designated bridge ID           : 32768.00:90:69:0b:47:d1
Port role                      : Designated
Link type                      : Pt-Pt/NONEDGE
Boundary port                   : No


Spanning tree interface parameters for instance 1

Interface name                 : ae1
Port identifier                : 128.1
Designated port ID             : 128.1
Port cost                      : 1000
Port state                     : Forwarding
Designated bridge ID           : 32768.00:90:69:0b:47:d1
```

```
Port role                       : Designated
Link type                       : Pt-Pt/NONEDGE
Boundary port                    : No

Interface name                  : ge-2/1/2
Port identifier                 : 128.2
Designated port ID              : 128.2
Port cost                       : 20000
Port state                      : Forwarding
Designated bridge ID            : 32768.00:90:69:0b:47:d1
Port role                       : Designated
Link type                       : Pt-Pt/NONEDGE
Boundary port                    : No

Interface name                  : ge-2/1/5
Port identifier                 : 128.3
Designated port ID              : 128.3
Port cost                       : 29999
Port state                      : Forwarding
Designated bridge ID            : 32768.00:90:69:0b:47:d1
Port role                       : Designated
Link type                       : Pt-Pt/NONEDGE
Boundary port                    : No

Interface name                  : ge-2/2/1
Port identifier                 : 128.4
Designated port ID              : 128.26
Port cost                       : 20000
Port state                      : Forwarding
Designated bridge ID            : 32768.00:13:c3:9e:c8:80
Port role                       : Root
Link type                       : Pt-Pt/NONEDGE
Boundary port                    : No

...
```

show spanning-tree interface msti

```
user@host> show spanning-tree interface msti 1 routing-instance vs1 detail
 Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
xe-7/0/0         128:1         128:1  32769.0090690b4fd1      2000  FWD    DESG
ge-5/1/0         128:2         128:2  32769.0090690b4fd1     20000  FWD    DESG
ge-5/1/1         128:3         128:3  32769.0090690b4fd1     20000  FWD    DESG
ae1              128:4         128:1  32769.0090690b47d1     10000  BLK    ALT
ge-5/1/4         128:5         128:3  32769.0090690b47d1     20000  BLK    ALT
xe-7/2/0         128:6         128:6  32769.0090690b47d1      2000  FWD    ROOT
```

show spanning-tree interface vlan-id

```
user@host> show spanning-tree interface vlan-id 101 routing-instance vs1 detail
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ge-11/0/5        128:1         128:1  32768.0090690b7fd1     20000  FWD    DESG
ge-11/0/6        128:2         128:1  32768.0090690b7fd1     20000  BLK    BKUP
ge-11/1/0        128:3         128:2  32768.0090690b4fd1     20000  BLK    ALT
ge-11/1/1        128:4         128:3  32768.0090690b4fd1     20000  BLK    ALT
```

```
ge-11/1/4        128:5        128:1  32768.0090690b47d1     20000  BLK    ALT
xe-10/0/0        128:6        128:5  32768.0090690b4fd1      2000  BLK    ALT
xe-10/2/0        128:7        128:4  32768.0090690b47d1      2000  FWD    ROOT
```

show spanning-tree interface (VSTP)

```
user@host> show spanning-tree interface
Spanning tree interface parameters for instance 0

Interface    Port ID    Designated      Designated         Cost    State  Role
                         port ID        bridge ID
ge-1/0/1         128:1        128:1  28672.0090690b3fe0     20000  FWD    DESG
ge-1/0/2         128:2        128:2  28672.0090690b3fe0     20000  FWD    DESG

Spanning tree interface parameters for VLAN 10

Interface    Port ID    Designated      Designated         Cost    State  Role
                         port ID        bridge ID
ge-1/0/1         128:1        128:1  28672.0090690b3fe0     20000  FWD    DESG
ge-1/0/2         128:2        128:2  28672.0090690b3fe0     20000  FWD    DESG

Spanning tree interface parameters for VLAN 20

Interface    Port ID    Designated      Designated         Cost    State  Role
                         port ID        bridge ID
ge-1/0/1         128:1        128:1  28672.0090690b3fe0     20000  FWD    DESG
ge-1/0/2         128:2        128:2  28672.0090690b3fe0     20000  FWD    DESG
```

show spanning-tree interface vlan-id (VSTP)

```
user@host> show spanning-tree interface vlan-id 10
Spanning tree interface parameters for VLAN 10

Interface    Port ID    Designated      Designated         Cost    State  Role
                         port ID        bridge ID
ge-1/0/1         128:1        128:1  28672.0090690b3fe0     20000  FWD    DESG
ge-1/0/2         128:2        128:2  28672.0090690b3fe0     20000  FWD    DESG
```

show spanning-tree mstp configuration

List of Syntax
Syntax on page 443
Syntax (EX Series Switch and the QFX Series) on page 443

Syntax
show spanning-tree mstp configuration
<brief | detail>
<routing-instance *routing-instance-name*>

Syntax (EX Series Switch and the QFX Series)
show spanning-tree mstp configuration
<brief | detail>

Release Information
Command introduced in Junos OS Release 8.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description
Display the MSTP configuration.

Options
none—Display MSTP configuration information.

brief | detail—(Optional) Display the specified level of output.

routing-instance *routing-instance-name*—(Optional) Display MSTP configuration information for the specified routing instance.

Required Privilege Level
view

List of Sample Output
show spanning-tree mstp configuration detail on page 444
show spanning-tree mstp configuration detail (QFX Series) on page 444

Output Fields
Table 46 on page 443 lists the output fields for the **show spanning-tree mstp configuration** command. Output fields are listed in the approximate order in which they appear.

Table 46: show spanning-tree mstp configuration Output Fields

Field Name	Field Description
Context id	Internally generated identifier.
Region name	MSTP region name carried in the MSTP BPDUs.
Revision	Revision number of the MSTP configuration.
Configuration digest	Numerical value derived from the VLAN-to-instance mapping table.
MSTI	MST instance identifier.
Member VLANs	VLAN identifiers associated with the MSTI.

Sample Output

show spanning-tree mstp configuration detail

```
user@host> show spanning-tree mstp configuration routing-instance vs1 detail
MSTP configuration information
Context identifier     : 1
Region name            : henry
Revision               : 3
Configuration digest   : 0x6da4b5c4fd587757eef35675365e1


MSTI     Member VLANs
   0 0-99,101-199,201-4094
   1 100
   2 200
```

show spanning-tree mstp configuration detail (QFX Series)

```
user@lf0> show spanning-tree mstp configuration routing-instance vs1 detail
MSTP configuration information
Context identifier     : 1
Region name            : henry
Revision               : 3
Configuration digest   : 0x6da4b5c4fd587757eef35675365e1


MSTI     Member VLANs
   0 0-99,101-199,201-4094
   1 100
   2 200
```

show spanning-tree statistics

List of Syntax
Syntax on page 445
Syntax (EX Series Switch and the QFX Series) on page 445

Syntax
show spanning-tree statistics
<brief | detail>
<interface *interface-name*>
<routing-instance *routing-instance-name*>

Syntax (EX Series Switch and the QFX Series)
show spanning-tree statistics
<brief | detail>
<interface *interface-name* | vlan *vlan-id*>

Release Information
Command introduced in Junos OS Release 8.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for QFX Series switches.

Description
Display STP statistics.

Options
none—Display brief STP statistics.

brief | detail—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display STP statistics for the specified interface.

routing-instance *routing-instance-name*—(Optional) Display STP statistics for the specified routing instance.

Required Privilege Level
view

List of Sample Output
show spanning-tree statistics routing-instance on page 446
show spanning-tree statistics interface routing-instance detail on page 446

Output Fields
Table 47 on page 445 lists the output fields for the **show spanning-tree statistics** command. Output fields are listed in the approximate order in which they appear.

Table 47: show spanning-tree statistics Output Fields

Field Name	Field Description
Message type	Type of message being counted.
BPDUs sent	Total number of BPDUs sent.
BPDUs received	Total number of BPDUs received.
BPDUs sent in last interval	Number of BPDUs sent within a specified interval.
BPDUs received in last interval	Number of BPDUs received within a specified interval.

Table 47: show spanning-tree statistics Output Fields *(continued)*

Field Name	Field Description
Interface	Interface for which the statistics are being displayed.
Next BPDU transmission	Number of seconds until the next BPDU is scheduled to be sent.

Sample Output

show spanning-tree statistics routing-instance

```
user@host> show spanning-tree statistics routing-instance vs1 detail
Routing instance level STP statistics
Message type                    : bpdus
BPDUs sent                      : 1396
BPDUs received                  : 1027
BPDUs sent in last interval     : 5         (duration: 4 sec)
BPDUs received in last interval: 4          (duration: 4 sec)
```

show spanning-tree statistics interface routing-instance detail

```
user@host> show spanning-tree statistics interface ge-11/1/4 routing-instance vs1 detail
Interface   BPDUs sent   BPDUs received      Next BPDU
                                            transmission
ge-11/1/4            7           190                0
```

show system statistics arp

Syntax show system statistics arp

Release Information Command introduced in Junos OS Release 9.6 for EX Series switches.

Description Display system-wide Address Resolution Protocol (ARP) statistics.

Required Privilege Level view

Related Documentation
- *Example: Configuring Proxy ARP on an EX Series Switch*
- Verifying That Proxy ARP Is Working Correctly on page 377

show system statistics arp

```
user@switch> show system statistics arp
arp:
        90060 datagrams received
        34 ARP requests received
        610 ARP replies received
        0 resolution request  received
        0 unrestricted proxy requests
        0 restricted proxy requests
        0 received proxy requests
        0 unrestricted proxy requests not proxied
        0 restricted proxy requests not proxied
        0 datagrams with bogus interface
        0 datagrams with incorrect length
        0 datagrams for non-IP protocol
        0 datagrams with unsupported op code
        0 datagrams with bad protocol address length
        0 datagrams with bad hardware address length
        0 datagrams with multicast source address
        0 datagrams with multicast target address
        0 datagrams with my own hardware address
        0 datagrams for an address not on the interface
        0 datagrams with a broadcast source address
        294 datagrams with source address duplicate to mine
        89113 datagrams which were not for me
        0 packets discarded waiting for resolution
        0 packets sent after waiting for resolution
        309 ARP requests sent
        35 ARP replies sent
        0 requests for memory denied
        0 requests dropped on entry
        0 requests dropped during retry
        0 requests dropped due to interface deletion
        0 requests on unnumbered interfaces
        0 new requests on unnumbered interfaces
        0 replies for from unnumbered interfaces
        0 requests on unnumbered interface with non-subnetted donor
        0 replies from unnumbered interface with non-subnetted donor
```

show vlans

Syntax

show vlans
<brief | detail | extensive>
<dot1q-tunneling>
<sort-by (tag | name)>
<vlan-range-name>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.
Option **dot1q-tunneling** added in Junos OS Release 12.1 for the QFX Series.

Description

Display information about VLANs configured on bridged Ethernet interfaces. For interfaces configured to support a VoIP VLAN and a data VLAN, the **show vlans** command displays both tagged and untagged membership for those VLANs.

NOTE: When a series of VLANs is created using the **vlan-range** statement, such VLAN names are preceded and followed by a double underscore. For example, a series of VLANs using the VLAN range 1 through 3 and the base VLAN name **marketing** would be displayed as **__marketing_1__**, **__marketing_2__**, and **__marketing_3__**.

NOTE: To display an 802.1X supplicant successfully authenticated in multiple-supplicant mode with dynamic VLAN movement, use the **show vlans** *vlan-name* **extensive** operational mode command, where *vlan-name* is the dynamic VLAN.

Options

none—Display information for all VLANs. VLAN information is displayed by VLAN name in ascending order.

brief | detail | extensive—(Optional) Display the specified level of output.

sort-by (tag | name)—(Optional) Display VLANs in ascending order of VLAN IDs or VLAN names.

vlan-range-name—(Optional) Display VLANs in ascending order of VLAN range names.

Required Privilege Level

view

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on the QFX Series on page 85
- Example: Setting Up Bridging with Multiple VLANs on page 102
- *Understanding Bridging*
- show ethernet-switching interfaces on page 388

List of Sample Output

Output Fields

Table 48 on page 449 lists the output fields for the **show vlans** command. Output fields are listed in the approximate order in which they appear.

Table 48: show vlans Output Fields

Field Name	Field Description	Level of Output
Name	Name of a VLAN.	none, **brief**
Tag	802.1Q tag applied to this VLAN. If **none** is displayed, no tag is applied.	All levels
Interfaces	Interface associated with learned MAC addresses or **All-members** option (flood entry). An asterisk (*) beside the interface indicates that the interface is **UP**.	All levels
Address	IP address.	none, **brief**
Ports Active /Total	Number of interfaces associated with a VLAN: **Active** indicates interfaces that are **UP**, and **Total** indicates interfaces that are active and inactive.	**brief**
VLAN	Name of a VLAN.	**detail**, **extensive**
Admin state	State of the interface. Values are: **enabled**—The interface is turned on, and the physical link is operational and can pass packets.	**detail**,**extensive**
MAC learning Status	Indicates if MAC learning is disabled.	**detail**, **extensive**
Description	Description for the VLAN.	**detail**,**extensive**
Primary IP	Primary IP address associated with a VLAN.	**detail**
Number of interfaces	Number of interfaces associated with a VLAN. Both the total number of interfaces and the number of active interfaces associated with a VLAN are displayed.	**detail**, **extensive**
STP	Spanning tree associated with a VLAN.	**detail**,**extensive**
Tagged interfaces	Tagged interfaces with which a VLAN is associated.	**detail**,**extensive**

Table 48: show vlans Output Fields *(continued)*

Field Name	Field Description	Level of Output
Untagged interfaces	Untagged interfaces with which a VLAN is associated.	**detail. extensive**
Dot1q Tunneling Status	Indicates if Q-in-Q tunneling is enabled.	**extensive**
Customer VLAN ranges	List of customer VLAN (C-VLAN) ranges associated with this service VLAN (S-VLAN).	**extensive**
Private VLAN Mode	The private VLAN mode for this VLAN. Values include **Primary**, **Isolated**, and **Community**.	**extensive**
Primary VLAN	Primary VLAN tag for this secondary VLAN.	**extensive**
Internal Index	VLAN index internal to Junos OS software.	**extensive**
Origin	Manner in which the VLAN was created: **static** or **learn**.	**extensive**
Protocol	Port-based VLAN or MAC-based VLAN. MAC-based protocol is displayed when VLAN assignment is done either statically or dynamically through 802.1X,	**extensive**
IP addresses	IP address associated with a VLAN.	**extensive**
Number of MAC entries	For MAC-based VLANs created either statically or dynamically, the MAC addresses associated with an interface.	**extensive**
Number of mapping rules	Number of mapping rules for Q-in-Q tunneling (**Push**) and VLAN translation (**Swap**).	
Secondary VLANs	Secondary VLANs associated with a primary VLAN.	**extensive**
Isolated VLANs	Isolated VLANs associated with a primary VLAN.	**extensive**
Community VLANs	Community VLANs associated with a primary VLAN.	**extensive**
VLANs summary	VLAN counts: • **Total**—Total number of VLANs on the switch. • **Configured VLANs**—Number of VLANs that are based on user-configured settings. • **Internal VLANs**—Number of VLANs created by the system with no explicit configuration or protocol—for example, the **default** VLAN and the VLAN created when a trunk interface is not configured with native VLAN membership. • **Temporary VLANs**—Number of VLANs from the previous configuration that the system retains for a limited time after restart. Temporary VLANs are converted into one of the other types of VLAN, or are removed from the system if the current configuration does not require them.	All levels

Table 48: show vlans Output Fields *(continued)*

Field Name	Field Description	Level of Output

Dot1q VLANs summary	802.1Q VLAN counts:
	• **Total**—Total number of 802.1Q-tagged and untagged VLANs on the switch.
	• **Tagged VLANs**—Number of 802.1Q-tagged VLANs.
	• **Untagged VLANs**—Number of untagged 802.1Q VLANs.
	• **Private VLAN**—Counts of the following kinds of 802.1Q private VLANs (PVLANs):
	  ◦ **Primary VLANs**—Number of primary forwarding private VLANs.
	  ◦ **Community VLANs**—Number of community transporting and forwarding private VLANs.
	  ◦ **Isolated VLANs**—Number of isolated receiving and forwarding private VLANs.
	  ◦ **Inter–switch–isolated VLANs**—Number of inter-switch isolated receiving and forwarding private VLANs.
		All levels

Dot1q Tunneled VLANs summary	Q-in-Q-tunneled VLAN counts: • **Total**—Total number of Q-in-Q-tunneled VLANs on the switch. • **Private VLAN**—Counts of primary, community, and isolated Q-in-Q-tunneled private VLANs (PVLANs).	All levels

Sample Output

show vlans

```
user@switch> show vlans

Name            Tag     Interfaces
default         None
                        xe-0/0/34.0, xe-0/0/33.0, xe-0/0/32.0, xe-0/0/31.0,
                        xe-0/0/30.0, xe-0/0/29.0, xe-0/0/28.0, xe-0/0/27.0,
                        xe-0/0/26.0, xe-0/0/25.0, xe-0/0/19.0, xe-0/0/18.0,
                        xe-0/0/17.0, xe-0/0/16.0, xe-0/0/15.0, xe-0/0/14.0,
                        xe-0/0/13.0, xe-0/0/11.0, xe-0/0/9.0, xe-0/0/8.0,
                        xe-0/0/3.0, xe-0/0/2.0, xe-0/0/1.0
v0001           1
                        xe-0/0/24.0, xe-0/0/23.0, xe-0/0/22.0, xe-0/0/21.0
v0002           2
                        None
v0003           3
                        None
v0004           4
                        None
v0005           5
                        None
```

show vlans (Private VLANs)

```
user@switch> show vlans

Name            Tag     Interfaces
__pvlan_pvlan_xe-0/0/46.0__
```

```
                                xe-0/0/44.0*, xe-0/0/46.0*
c1
                                xe-0/0/4.0*, xe-0/0/44.0*
c2
                                xe-0/0/28.0*, xe-0/0/44.0*
default
                                None
pvlan           500
                                xe-0/0/4.0*, xe-0/0/28.0*, xe-0/0/44.0*, xe-0/0/46.0*
```

show vlans brief

```
user@switch> show vlans brief
                                                Ports
Name            Tag     Address                 Active/Total
default         None                            0/23
v0001           1                               0/4
v0002           2                               0/0
v0003           3                               0/0
v0004           4                               0/0
v0005           5                               0/0
v0006           6                               0/0
v0007           7                               0/0
v0008           8                               0/0
v0009           9                               0/0
v0010           10                              0/2
v0011           11                              0/0
v0012           12                              0/0
v0013           13                              0/0
v0014           14                              0/0
v0015           15                              0/0
v0016           16                              0/0
```

show vlans detail

```
user@switch> show vlans detail
VLAN: default, Tag: Untagged, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 23 (Active = 0)
  STP: None, RTG: None
  Untagged interfaces: xe-0/0/34.0, xe-0/0/33.0, xe-0/0/32.0, xe-0/0/31.0,
  xe-0/0/30.0, xe-0/0/29.0, xe-0/0/28.0, xe-0/0/27.0, xe-0/0/26.0,
  xe-0/0/25.0, xe-0/0/19.0, xe-0/0/18.0, xe-0/0/17.0, xe-0/0/16.0,
  xe-0/0/15.0, xe-0/0/14.0, xe-0/0/13.0, xe-0/0/11.0, xe-0/0/9.0, xe-0/0/8.0,
  xe-0/0/3.0, xe-0/0/2.0, xe-0/0/1.0,
  Tagged interfaces: None

VLAN: v0001, Tag: 802.1Q Tag 1, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 4 (Active = 0)
  Dot1q Tunneling Status: Enabled
  STP: None, RTG: None
  Untagged interfaces: None
  Tagged interfaces: xe-0/0/24.0, xe-0/0/23.0, xe-0/0/22.0, xe-0/0/21.0,

VLAN: v0002, Tag: 802.1Q Tag 2, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 0 (Active = 0)
  STP: None, RTG: None
  Untagged interfaces: None
  Tagged interfaces: None
```

```
VLAN: v0003, Tag: 802.1Q Tag 3, Admin state: Enabled
  Description: None
  Primary IP: None, Number of interfaces: 0 (Active = 0)
  STP: None, RTG: None
  Untagged interfaces: None
  Tagged interfaces: None

VLAN: vlan4000, 802.1Q Tag: Untagged, Admin State: Enabled
MAC learning Status: Disabled
Number of interfaces: 0 (Active = 0)
```

show vlans extensive (Port-Based)

```
user@switch> show vlans extensive
VLAN: default, created at Mon Feb  4 12:13:47 2008
  Tag: None, Internal index: 0, Admin state: Enabled, Origin: static
  Description: None
  Customer VLAN ranges:
                    1-4100
    Protocol: Port based
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 0 (Active = 0), Untagged 23 (Active = 0)
    xe-0/0/34.0 (untagged, access)
    xe-0/0/33.0 (untagged, access)
    xe-0/0/32.0 (untagged, access)
    xe-0/0/31.0 (untagged, access)
    xe-0/0/30.0 (untagged, access)
    xe-0/0/29.0 (untagged, access)
    xe-0/0/28.0 (untagged, access)
    xe-0/0/27.0 (untagged, access)
    xe-0/0/26.0 (untagged, access)
    xe-0/0/25.0 (untagged, access)
    xe-0/0/19.0 (untagged, access)
    xe-0/0/18.0 (untagged, access)
    xe-0/0/17.0 (untagged, access)
    xe-0/0/16.0 (untagged, access)
    xe-0/0/15.0 (untagged, access)
    xe-0/0/14.0 (untagged, access)
    xe-0/0/13.0 (untagged, access)
    xe-0/0/11.0 (untagged, access)
    xe-0/0/9.0 (untagged, access)
    xe-0/0/8.0 (untagged, access)
    xe-0/0/3.0 (untagged, access)
    xe-0/0/2.0 (untagged, access)
    xe-0/0/1.0 (untagged, access)

Secondary VLANs: Isolated 1, Community  1
  Isolated VLANs :
      __pvlan_pvlan_xe-0/0/3.0__
  Community VLANs :
      comm1

VLAN: v0001, created at Mon Feb  4 12:13:47 2008
  Tag: 1, Internal index: 1, Admin state: Enabled, Origin: static
  Description: None
  Protocol: Port based, Layer 3 interface: None
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 4 (Active = 0), Untagged 0 (Active = 0)
```

```
    xe-0/0/24.0 (tagged, trunk)
    xe-0/0/23.0 (tagged, trunk)
    xe-0/0/22.0 (tagged, trunk)
    xe-0/0/21.0 (tagged, trunk)

VLAN: v0002, created at Mon Feb  4 12:13:47 2008
  Tag: 2, Internal index: 2, Admin state: Enabled, Origin: static
  Description: None
  Protocol: Port based, Layer 3 interface: None
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 0 (Active = 0), Untagged 0 (Active = 0)
    None

VLAN: v0003, created at Mon Feb  4 12:13:47 2008
  Tag: 3, Internal index: 3, Admin state: Enabled, Origin: static
  Description: None
  Protocol: Port based, Layer 3 interface: None
  IP addresses: None
  STP: None, RTG: None.
  Number of interfaces: Tagged 0 (Active = 0), Untagged 0 (Active = 0)
    None
```

show vlans (Q-in-Q Tunneling)

```
user@switch> show vlans dot1q-tunneling
Name           Tag     Interfaces
sv100          100     xe-0/0/4.0*, xe-0/0/15.0*
```

show vlans extensive (Q-in-Q Tunneling)

```
user@switch> show vlans sv100 extensive
VLAN: sv100, Created at: Sat Sep 10 12:53:52 2011
802.1Q Tag: 100, Internal index: 2, Admin State: Enabled, Origin: Static
Dot1q Tunneling Status: Enabled
Customer VLAN ranges:
                    10-20
                    40-50
Protocol: Port Mode
Number of interfaces: Tagged 1 (Active = 1), Untagged  0 (Active = 0)
      ge-0/0/0.0, tagged, trunk

Number of mapping rules:
      Push 1 (Active = 0), Policy 0 (Active = 0), Swap 0 (Active = 0)

      xe-0/0/3.0*, 300, push
```

show vlans extensive (Q-in-Q Tunneling and L2TP)

```
user@switch> show vlans v1 extensive
VLAN: v1, Created at: Fri Mar 2 05:07:38 2012
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Dot1q Tunneling status: Enabled
Layer2 Protocol Tunneling status: Enabled
```

show vlans sort-by tag

```
user@switch> show vlans sort-by tag
Name           Tag     Interfaces
default                        None
__vlan-x_1__   1
```

```
                                None
__vlan-x_2__    2
                                None
__vlan-x_3__    3
                                None
__vlan-x_4__    4
                                None
__vlan-x_5__    5
                                None
__vlan-x_6__    6
                                None
__vlan-x_7__    7
                                None
__vlan-x_8__    8
                                None
__vlan-x_9__    9
                                None
__vlan-x_10__   10
                                None
__vlan-x_11__   11
                                None
__vlan-x_12__   12
                                None
__vlan-x_13__   13
                                None
__vlan-x_14__   14
                                None
__vlan-x_15__   15
                                None
__vlan-x_16__   16
                                None
__vlan-x_17__   17
                                None
__vlan-x_18__   18
                                None
__vlan-x_19__   19
                                None
__vlan-x_20__   20
                                None
```

show vlans sort-by name

```
user@switch> show vlans sort-by employee

Name              Tag   Interfaces

__employee_120__ 120
                        xe-0/0/22.0*
__employee_121__ 121
                        xe-0/0/22.0*
__employee_122__ 122
                        xe-0/0/22.0*
__employee_123__ 123
                        xe-0/0/22.0*
__employee_124__ 124
                        xe-0/0/22.0*
__employee_125__ 125
                        xe-0/0/22.0*
__employee_126__ 126
                        xe-0/0/22.0*
__employee_127__ 127
```

```
                                xe-0/0/22.0*
__employee_128__ 128
                                xe-0/0/22.0*
__employee_129__ 129
                                xe-0/0/22.0*
__employee_130__ 130
                                xe-0/0/22.0*
```

show vlans tag

```
user@switch> show vlans employee

Name             Tag   Interfaces

__employee_120__ 120
                                xe-0/0/22.0*
__employee_121__ 121
                                xe-0/0/22.0*
__employee_122__ 122
                                xe-0/0/22.0*
__employee_123__ 123
                                xe-0/0/22.0*
__employee_124__ 124
                                xe-0/0/22.0*
__employee_125__ 125
                                xe-0/0/22.0*
__employee_126__ 126
                                xe-0/0/22.0*
__employee_127__ 127
                                xe-0/0/22.0*
__employee_128__ 128
                                xe-0/0/22.0*
__employee_129__ 129
                                xe-0/0/22.0*
__employee_130__ 130
                                xe-0/0/22.0*
```

PART 4

Troubleshooting

CHAPTER 43

Troubleshooting Procedures

Troubleshooting Ethernet Switching

Problem **Description:** Sometimes a MAC address entry in the switch's Ethernet switching table is not updated after the device with that MAC address has been moved from one interface to another on the switch. Typically, the switch does not wait for a MAC address expiration when a MAC move operation occurs. As soon as the switch detects the MAC address on the new interface, it immediately updates the table. Many network devices send a gratuitous ARP packet when switching an IP address from one device to another. The switch updates its ARP cache table after receipt of such gratuitous ARP messages, and then it also updates its Ethernet switching table.

Sometimes silent devices, such as syslog servers or SNMP trap receivers that receive UDP traffic but do not return acknowledgment (ACK) messages to the traffic source, fail to send gratuitous ARP packets when a device moves. If such a move occurs when the system administrator is not available to explicitly clear the affected interfaces by issuing the **clear ethernet-switching table** command, the entry for the moved device in the Ethernet switching table is not updated.

Solution Set up the switch to handle unattended MAC address switchovers.

1. Reduce the system-wide ARP aging timer. (By default, the ARP aging timer is set at 20 minutes. The range of the ARP aging timer is from 1 through 240 minutes.)

   ```
   [edit system arp]
   user@switch# set aging-timer 3
   ```

2. Set the MAC aging timer to the same value as the ARP timer. (By default, the MAC aging timer is set to 300 seconds. The range is 15 to 1,000,000 seconds.)

   ```
   [edit vlans]
   user@switch# set vlans sales mac-table-aging-time 180
   ```

The ARP entry and the MAC address entry for the moved device expire within the times specified by the aging timer values. After the entries expire, the switch sends a new ARP message to the IP address of the device. The device responds to the ARP message, thereby refreshing the entries in the switch's ARP cache table and Ethernet switching table.

Related Documentation

- arp on page 350
- mac-table-aging-time on page 285