Ethernet Switching Feature Guide

Modified: 2018-09-06

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

Ethernet Switching Feature Guide

END USER LICENSE AGREEMENT

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 https://www.juniper.net/documentation/.

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 https://www.juniper.net/books.

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 CLI Explorer.

Documentation Conventions

Table 1 on page xxxiii 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 xxxiii 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**.

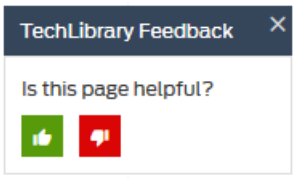
Table 2: Text and Syntax Conventions (continued)

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 so that we can improve our documentation. You can use either of the following methods:

- Online feedback system—Click TechLibrary Feedback, on the lower right of any page on the Juniper Networks TechLibrary site, and do one of the following:



  - Click the thumbs-up icon if the information on the page was helpful to you.
  - Click the thumbs-down icon if the information on the page was not helpful to you or if you have suggestions for improvement, and use the pop-up form to provide 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 https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://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: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: https://www.juniper.net/cm/

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

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 https://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 https://www.juniper.net/support/requesting-support.html.

PART 1

Configuring Ethernet Switching

CHAPTER 1

Layer 2 Networking Overview

Using the Enhanced Layer 2 Software CLI

Enhanced Layer 2 Software (ELS) provides a uniform CLI for configuring and monitoring Layer 2 features on QFX Series switches, EX Series switches, and other Juniper Networks devices, such as MX Series routers. With ELS, you configure Layer 2 features in the same way on all these Juniper Networks devices.

This topic explains how to know if your platform is running ELS. It also explains how to perform some common tasks using the ELS style of configuration.

Understanding Which Devices Support ELS

ELS is automatically supported if your device is running a Junos OS release that supports it. You do not need to take any action to enable ELS, and you cannot disable ELS. See Feature Explorer for information about which platforms and releases support ELS.

Understanding How to Configure Layer 2 Features Using ELS

Because ELS provides a uniform CLI, you can now perform the following tasks on supported devices in the same way:

Configuring a VLAN

You can configure one or more VLANs to perform Layer 2 bridging. The Layer 2 bridging functions include integrated routing and bridging (IRB) for support for Layer 2 bridging and Layer 3 IP routing on the same interface. EX Series and QFX Series switches can function as Layer 2 switches, each with multiple bridging, or broadcast, domains that participate in the same Layer 2 network. You can also configure Layer 3 routing support for a VLAN.

To configure a VLAN:

1. Create the VLAN by setting a unique VLAN name and configuring the VLAN ID:

   ```
   [edit]
   user@host#  set vlans vlan-name vlan-id vlan-id-number
   ```

   Using the VLAN ID list option, you can optionally specify a range of VLAN IDs.

   ```
   [edit]
   user@host#  set vlans vlan-name vlan-id-list vlan-ids | vlan-id--vlan-id
   ```

2. Assign at least one interface to the VLAN:

   ```
   [edit]
   user@host#  set interface interface-name family ethernet-switching vlan members vlan-name
   ```

Configuring the Native VLAN Identifier

EX Series and QFX Series switches support receiving and forwarding routed or bridged Ethernet frames with 802.1Q VLAN tags. Typically, trunk ports, which connect switches to each other, accept untagged control packets, but do not accept untagged data packets. You can enable a trunk port to accept untagged data packets by configuring a native VLAN ID on the interface on which you want the untagged data packets to be received.

To configure the native VLAN ID:

1. On the interface on which you want untagged data packets to be received, set the interface mode to **trunk**, which specifies that the interface is in multiple VLANs and can multiplex traffic between different VLANs.

   ```
   [edit interfaces]
   user@host#  set interface-name unit logical-unit-number family ethernet-switching
   interface-mode trunk
   ```

2. Configure the native VLAN ID and assign the interface to the native VLAN ID:

   ```
   [edit interfaces]
   user@host#  set interface-name native-vlan-id number
   ```

3. Assign the interface to the native VLAN ID:

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family ethernet-switching vlan
members native-vlan-id-number
```

Configuring Layer 2 Interfaces

To ensure that your high-traffic network is tuned for optimal performance, explicitly configure some settings on the switch's network interfaces.

To configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface as a **trunk** interface:

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family ethernet-switching
interface-mode trunk
```

To configure a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface as a **access** interface:

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family ethernet-switching
interface-mode access
```

To assign an interface to VLAN:

```
[edit interfaces]
user@host# set interface-name unit logical-unit-number family ethernet-switching vlan members
[all | vlan-names | vlan-ids]
```

Configuring Layer 3 Interfaces

To configure a Layer 3 interface, you must assign an IP address to the interface. You assign an address to an interface by specifying the address when you configure the protocol family. For the **inet** or **inet6** family, configure the interface IP address.

You can configure 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, 192.168.1.1). An IPv4 address with destination prefix utilizes a 4-octet dotted decimal address syntax with a destination prefix appended (for example, 192.168.1.1/16).

To specify an IP4 address for the logical unit:

```
[edit]
user@host# set interfaces interface-name unit logical-unit-number family inet address ip-address
```

You represent IP version 6 (IPv6) addresses in hexadecimal notation by using a colon-separated list of 16-bit values. You assign a 128-bit IPv6 address to an interface.

To specify an IP6 address for the logical unit:

```
[edit]
```

```
user@host# set interfaces interface-name unit logical-unit-number family inet6 address ip-address
```

Configuring an IRB Interface

Integrated routing and bridging (IRB) provides support for Layer 2 bridging and Layer 3 IP routing on the same interface. IRB enables you to route packets to another routed interface or to another VLAN that has a Layer 3 protocol configured. IRB interfaces enable the device to recognize packets that are being sent to local addresses so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated. An interface named irb functions as a logical router on which you can configure a Layer 3 logical interface for VLAN. For redundancy, you can combine an IRB interface with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments.

To configure an IRB interface:

1. Create a Layer 2 VLAN by assigning it a name and a VLAN ID:

   ```
   [edit]
   user@host# set vlans vlan-name vlan-id vlan-id
   ```

2. Create an IRB logical interface:

   ```
   [edit]
   user@host# set interface irb unit logical-unit-number family inet address ip-address
   ```

3. Associate the IRB interface with the VLAN:

   ```
   [edit]
   user@host# set vlans vlan-name l3-interface irb.logical-unit-number
   ```

Configuring an Aggregated Ethernet Interface and Configuring LACP on That Interface

Use the link aggregation feature to aggregate one or more links to form a virtual link or link aggregation group (LAG). The MAC client can treat this virtual link as if it were a single link to increase bandwidth, provide graceful degradation as failure occurs, and increase availability.

To configure an aggregated Ethernet interface:

1. Specify the number of aggregated Ethernet interfaces to be created:

   ```
   [edit chassis]
   user@host# set aggregated-devices ethernet device-count number
   ```

2. Specify the name of the link aggregation group interface:

   ```
   [edit]
   user@host# set interfaces aex
   ```

3. Specify the minimum number of links for the aggregated Ethernet interface (aex)—that is, the defined bundle— to be labeled *up*:

```
[edit interfaces]
user@host# set aex aggregated-ether-options minimum-links number
```

4. Specify the link speed for the aggregated Ethernet bundle:

```
[edit interfaces]
user@host# set aex aggregated-ether-options link-speed link-speed
```

5. Specify the members to be included within the aggregated Ethernet bundle:

```
[edit interfaces]
user@host# set interface-name ether-options 802.3ad aex
user@host# set interface-name ether-options 802.3ad aex
```

6. Specify an interface family for the aggregated Ethernet bundle:

```
[edit interfaces]
user@host# set aex unit 0 family inet address ip-address
```

For aggregated Ethernet interfaces on the device, you can configure the Link Aggregation Control Protocol (LACP). LACP bundles several physical interfaces to form one logical interface. You can configure aggregated Ethernet with or without LACP enabled.

When LACP is enabled, the local and remote sides of the aggregated Ethernet links exchange protocol data units (PDUs), containing information about the state of the link. You can configure Ethernet links to actively transmit PDUs, or you can configure the links to passively transmit them, sending out LACP PDUs only when they receive them from another link. One side of the link must be configured as active for the link to be up.

To configure LACP:

1. Enable one side of the aggregated Ethernet link as active:

```
[edit interfaces]
user@host# set aex aggregated-ether-options lacp active
```

2. Specify the interval at which the interfaces send LACP packets:

```
[edit interfaces]
user@host# set aex aggregated-ether-options lacp periodic interval
```

Understanding ELS Configuration Statement and Command Changes

ELS was introduced in Junos OS Release 12.3R2 for EX9200 switches. ELS changes the CLI for some of the Layer 2 features on supported EX Series and QFX Series switches.

The following sections provide a list of existing commands that were moved to new hierarchy levels or changed on EX Series switches as part of this CLI enhancement effort. These sections are provided as a high-level reference only. For detailed information about

these commands, use the links to the configuration statements provided or see the technical documentation.

Changes to the ethernet-switching-options Hierarchy Level

This section outlines the changes to the **ethernet-switching-options** hierarchy level.

NOTE: The **ethernet-switching-options** hierarchy level has been renamed as **switch-options**.

Table 3: Renaming the ethernet-switching-options hierarchy

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  authentication-whitelist {
  ...
  }
}
	switch-options {
	  ...
	  authentication-whitelist {
	  ...
	  }
	}

ethernet-switching-options {
  interfaces *interface-name* {
  no-mac-learning;
  ...
  }
}
	switch-options {
	  interfaces *interface-name* {
	  no-mac-learning;
	  ...
	  }
	}

ethernet-switching-options {
  unknown-unicast-forwarding {
  (...)
  }
}
	switch-options {
	  unknown-unicast-forwarding {
	  (...)
	  }
	}

Table 3: Renaming the ethernet-switching-options hierarchy (continued)

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  voip {
  interface (all | [*interface-name* | access-ports]) {
  forwarding-class (assured-forwarding | best-effort |
  expedited-forwarding | network-control);
  vlan *vlan-name*;
  ...
  }
  }
}
	switch-options {
	  voip {
	  interface (all | [*interface-name* | access-ports]) {
	  forwarding-class (assured-forwarding | best-effort
	  | expedited-forwarding | network-control);
	  vlan *vlan-name*;
	  ...
	  }
	  }
	}

Table 4: RTG Statements

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  redundant-trunk-group {
  group *name* {
  description;
  interface *interface-name* {
  primary;
  }
  preempt-cutover-timer *seconds*;
  ...
  }
  }
}
	switch-options {
	  redundant-trunk-group {
	  group *name* {
	  description;
	  interface *interface-name* {
	  primary;
	  }
	  preempt-cutover-timer *seconds*;
	  ...
	  }
	  }
	}

Table 5: Deleted Statements

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  mac-notification {
  notification-interval *seconds*;
  ...
  }
}
	The statements have been removed from the **switch-options** hierarchy.

ethernet-switching-options {
  traceoptions {
  file *filename* <files *number*> <no-stamp> <replace>
  <size *size*> <world-readable | no-world-readable>;
  flag *flag* <disable>;
  ...
  }
}
	The statements have been removed from the **switch-options** hierarchy.

ethernet-switching-options {
  port-error-disable {
  disable-timeout *timeout*;
  ...
  }
}
	NOTE: The **port-error-disable** statement has been replaced with a new statement.
	
	interfaces *interface-name* family **ethernet-switching** {
	  recovery-timeout *seconds*;
	}

Changes to the Port Mirroring Hierarchy Level

NOTE: Statements have moved from the **ethernet-switching-options** hierarchy level to the **forwarding-options** hierarchy level.

Table 6: Port Mirroring hierarchy

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  analyzer (Port Mirroring) {
  name {
  ...
  }
  }
}
	forwarding-options {
	  analyzer (Port Mirroring) {
	  name {
	  ...
	  }
	  }
	}

Changes to the Layer 2 Control Protocol Hierarchy Level

The Layer 2 control protocol statements have moved from the **ethernet-switching-options** hierarchy to the **protocols** hierarchy.

Table 7: Layer 2 Control Protocol

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  bpdu-block {
  ...
  }
}
	protocols {
	  layer2-control {
	  bpdu-block {
	  ...
	  }
	  }
	}

Changes to the dot1q-tunneling Statement

The **dot1q-tunneling** statement has been replaced with a new statement and moved to a different hierarchy level.

Table 8: dot1q-tunneling

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  dot1q-tunneling {
  ether-type (0x8100 | 0x88a8 | 0x9100);
  ...
  }
}
	interfaces *interface-name* {
	  ether-options {
	  ethernet-switch-profile {
	  tag-protocol-id [*tpids*];
	  }
	  }
	}
	
	interfaces *interface-name* {
	  aggregated-ether-options {
	  ethernet-switch-profile {
	  tag-protocol-id [*tpids*];
	  }
	  }
	}

Changes to the L2 Learning Protocol

The **mac-table-aging-time** statement has been replaced with a new statement and moved to a different hierarchy level.

Table 9: mac-table-aging-time statement

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  mac-table-aging-time *seconds*;
  ...
}
	protocols {
	  l2-learning {
	  global-mac-table-aging-time *seconds*;
	  ...
	  }
	}

Changes to Nonstop Bridging

The **nonstop-bridging** statement has moved to a different hierarchy level.

Table 10: Nonstop Bridging statement

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  nonstop-bridging;
}
	protocols {
	  layer2-control {
	  nonstop-bridging {
	  }
	  }
	}

Changes to Port Security and DHCP Snooping

Port security and DHCP snooping statements have moved to different hierarchy levels.

NOTE: The statement **examine-dhcp** does not exist in the changed hierarchy. DHCP snooping is now enabled automatically when other DHCP security features are enabled on a VLAN. See *Configuring Port Security Features* for additional information.

Table 11: Port Security statements

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
secure-access-port {
interface (all | *interface-name*) {
(**dhcp-trusted** | no-dhcp-trusted);
static-ip *ip-address* {
mac *mac-address*;
vlan *vlan-name*;
}
}
vlan (all | *vlan-name*) {
(**arp-inspection** | no-arp-inspection);
dhcp-option82 {
disable;
circuit-id {
prefix *hostname*;
use-interface-description;
use-vlan-id;
}
remote-id {
prefix (*hostname* | mac | none);
use-interface-description;
use-string *string*;
}
vendor-id [*string*];
}
(**examine-dhcp** | no-examine-dhcp);
}
(**ip-source-guard** | no-ip-source-guard);
}
}
	vlans *vlan-name* forwarding-options{
	dhcp-security {
	arp-inspection;
	group *group-name* {
	interface*interface-name* {
	static-ip *ip-address* {
	mac *mac-address*;
	}
	}
	overrides {
	no-option82;
	trusted;
	}
	}
	ip-source-guard;
	no-dhcp-snooping;
	option-82 {
	circuit-id {
	prefix {
	host-name;
	routing-instance-name;
	}
	use-interface-description (device | logical);
	use-vlan-id;
	}
	remote-id {
	host-name;
	use-interface-description (device | logical);
	use-string *string*;
	}
	vendor-id {
	use-string *string*;
	}
	}
	}

TIP: For allowed mac configuration, the original hierarchy statement **set ethernet-switching-options secure-access-port interface ge-0/0/2 allowed-mac 00:05:85:3A:82:8** is replaced by the ELS command **set interfaces ge-0/0/2 unit 0 accept-source-mac mac-address 00:05:85:3A:82:8**

NOTE: DHCP snooping statements have moved to a different hierarchy level.

Table 12: DHCP Snooping Statements

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  secure-access-port {
  dhcp-snooping-file {
  location *local_pathname* | *remote_URL*;
  timeout *seconds*;
  write-interval *seconds*;
  }
	system [
	  processes [
	  dhcp-service
	  dhcp-snooping-file *local_pathname* | *remote_URL*;
	  write-interval *interval*;
	  }
	}

Changes to Configuring VLANs

The statements for configuring VLANs have moved to a different hierarchy level.

NOTE: Starting with Junos OS Release 14.1X53-D10 for EX4300 and EX4600 switches, when enabling *x*STP, you can enable it on some or all interfaces included in a VLAN. For example, if you configure VLAN 100 to include interfaces ge-0/0/0, ge-0/0/1, and ge-0/0/2, and you want to enable MSTP on interfaces ge-0/0/0 and ge-0/0/2, you can specify the **set protocols mstp interface ge-0/0/0** and **set protocols mstp interface ge-0/0/2** commands. In this example, you did not explicitly enable MSTP on interface ge-0/0/1; therefore, MSTP is not enabled on this interface.

Table 13: VLAN hierarchy

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  secure-access-port vlan (all | *vlan-name*{
  mac-move-limit
  }
	vlans *vlan-name* switch-options {
	  mac-move-limit
	}

ethernet-switching-options {
  static {
  vlan *vlan-id* {
  mac *mac-address* next-hop *interface-name*;
  ...
  }
  }
}
	NOTE: Statement is replaced with a new statement and has moved to a different hierarchy level.
	
	vlans {
	  vlan-name {
	  switch-options {
	  interface *interface-name* {
	  static-mac *mac-address*;
	  ...
	  }
	  }
	  }
	}

Table 13: VLAN hierarchy (continued)

Original Hierarchy	Changed Hierarchy

vlans {
  vlan-name {
  interface *interface-name* {
  egress;
  ingress;
  mapping (native (push | swap) | policy | *tag* (push | swap));
  pvlan-trunk;
  ...
  }
  }
}
	These statements have been removed. You can assign interfaces to a VLAN using the **[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching vlan members *vlan-name*]** hierarchy.

vlans {
  vlan-name {
  isolation-id *id-number*;
  ...
  }
}
	Statements have been removed.

vlans {
  vlan-name {
  interface **vlan**.*logical-interface-number*;
  ...
  }
}
	NOTE: Syntax is changed.
	
	vlans {
	  vlan-name {
	  interface **irb**.*logical-interface-number*;
	  ...
	  }
	}

vlans {
  vlan-name {
  l3-interface-ingress-counting *layer-3-interface-name*;
  ...
  }
}
	Statement is removed. Ingress traffic is automatically tracked.

vlans {
  vlan-name {
  no-local-switching;
  ...
  }
}
	Statement is removed.

vlans {
  vlan-name {
  no-mac-learning;
  ...
  }
}
	Statement has been moved to different hierarchy.
	
	vlans {
	  vlan-name {
	  switch-options {
	  no-mac-learning *limit*
	  ...
	  }
	  }
	}

Table 13: VLAN hierarchy (continued)

Original Hierarchy	Changed Hierarchy

vlans {
  vlan-name {
  primary-vlan *vlan-name*;
  ...
  }
}
	Statement has been removed.

vlans {
  vlan-name {
  vlan-prune;
  ...
  }
}
	Statement is removed.

vlans {
  vlan-name {
  vlan-range *vlan-id-low-vlan-id-high*;
  ...
  }
}
	NOTE: Statement has been replaced with a new statement.
	
	vlans {
	  vlan-name {
	  vlan-id-list [*vlan-id-numbers*];
	  ...
	  }
	}

vlans {
  vlan-name {
  l3-interface vlan.*logical-interface-number*;
  ...
  }
}
	NOTE: Syntax is changed.
	
	vlans {
	  vlan-name {
	  interface irb.*logical-interface-number*;
	  ...
	  }
	}

Table 14: Statements Moved to a Different Hierarchy

Original Hierarchy	Changed Hierarchy

vlans {
  vlan-name {
  dot1q-tunneling {
  customer-vlans (*id* | native | *range*);
  layer2-protocol-tunneling all | *protocol-name* {
  drop-threshold *number*;
  shutdown-threshold *number*;
  ...
  }
  }
  }
}
	interface *interface-name* {
	  encapsulation extended-vlan-bridge;
	  flexible-vlan-tagging;
	  native-vlan-id *number*;
	  unit *logical-unit-number* {
	  input-vlan-map *action*;
	  output-vlan-map *action*;
	  vlan-id *number*;
	  vlan-id-list [*vlan-id vlan-id–vlan-id*];
	  }
	}

Table 14: Statements Moved to a Different Hierarchy (continued)

Original Hierarchy	Changed Hierarchy

vlans {
  vlan-name {
  filter{
  input *filter-name*
  output *filter-name*;
  ...
  }
  }
}
	vlans {
	  vlan-name {
	  forwarding-options {
	  filter{
	  input *filter-name*
	  output *filter-name*;
	  ...
	  }
	  }
	  }
	}

vlans {
  vlan-name {
  mac-limit *limit* action *action*;
  ...
  }
}
	vlans {
	  vlan-name {
	  switch-options {
	  interface-mac-limit *limit* {
	  packet-action *action*;
	  ...
	  }
	  }
	  }
	}
	
	vlans {
	  vlan-name {
	  switch-options {
	  interface *interface-name* {
	  interface-mac-limit *limit* {
	  packet-action *action*;
	  ...
	  }
	  }
	  }
	  }
	}

vlans {
  vlan-name {
  mac-table-aging-time *seconds*;
  ...
  }
}
	protocols {
	  l2-learning {
	  global-mac-table-aging-time *seconds*;
	  ...
	  }
	}

Changes to Storm Control Profiles

Storm control is configured in two steps. The first step is to create a storm control profile at the **[edit forwarding-options]** hierarchy level, and the second step is to bind the profile to a logical interface at the **[edit interfaces]** hierarchy level. See *Example: Configuring Storm Control to Prevent Network Outages on EX Series Switches* for the changed procedure.

Table 15: Changes to the Storm Control Profile hierarchy level

Original Hierarchy	Changed Hierarchy

ethernet-switching-options {
  storm-control {
  (...)
  }
}
	forwarding-options {
	  storm-control-profiles *profile-name* {
	  (...)
	  }
	  }
	
	interfaces *interface-name* unit *number* family
	  ethernet-switching {
	  storm-control *storm-control-profile*;
	}

Changes to the Interfaces Hierarchy

NOTE: Statements have been moved to a different hierarchy.

Table 16: Changes to the Interfaces hierarchy

Original Hierarchy	Changed Hierarchy

interfaces *interface-name* {
  ether-options {
  link-mode *mode*;
  speed (auto-negotiation | *speed*)
  }
}
	interfaces *interface-name* {
	  link-mode *mode*;
	  speed *speed*)
	}

interfaces *interface-name* {
  unit *logical-unit-number* {
  family ethernet-switching {
  native-vlan-id *vlan-id*
  }
  }
}
	interfaces *interface-name* {
	  native-vlan-id *vlan-id*
	}

interfaces *interface-name* {
  unit *logical-unit-number* {
  family ethernet-switching {
  port-mode *mode*
  }
  }
}
	NOTE: Statement has been replaced with a new statement.
	
	interfaces *interface-name* {
	  unit *logical-unit-number* {
	  family ethernet-switching {
	  interface-mode *mode*
	  }
	  }
	}

interfaces vlan	NOTE: Statement has been replaced with a new statement. interfaces irb

Changes to IGMP Snooping

Table 17: IGMP Snooping hierarchy

Original Hierarchy	Changed Hierarchy

protocols {
  igmp-snooping {
  traceoptions {
  file *filename* <files *number*> <no-stamp> <replace>
  <size *maximum-file-size*> <world-readable |
  no-world-readable>;
  flag *flag* <*flag-modifier*> <disable>;
  }
  vlan (all | *vlan-identifier*) {
  disable;
  data-forwarding {
  receiver {
  install;
  source-vlans *vlan-name*;
  }
  source {
  groups *ip-address*;
  }
  }
  immediate-leave;
  interface (all | *interface-name*) {
  multicast-router-interface;
  static {
  group *multicast-ip-address*;
  }
  }
  proxy {
  source-address *ip-address*;
  }
  robust-count *number*;
  }
  }
}
	protocols {
	  igmp-snooping {
	  vlan *vlan-name* {
	  immediate-leave;
	  interface *(all | interface-name)* {
	  group-limit <1..65535>
	  host-only-interface
	  multicast-router-interface;
	  immediate-leave;
	  static {
	  group *multicast-ip-address* {
	  source < >
	  }
	  }
	  }
	  }
	  l2-querier {
	  source-address *ip-address*;
	  }
	  proxy {
	  source-address *ip-address*;
	  }
	  query-interval *number*;
	  query-last-member-interval *number*;
	  query-response-interval *number*;
	  robust-count *number*;
	  traceoptions {
	  file *filename* <files *number*> <no-stamp> <replace>
	  <size *maximum-file-size*> <world-readable |
	  no-world-readable>;
	  flag *flag* <*flag-modifier*>;
	  }
	  }
	  }
	}

Understanding the ELS Translator

ELS Translator is a Web-based tool that converts Junos OS Layer 2 configurations to Enhanced Layer 2 Software (ELS) configurations. This conversion tool supports all Juniper Networks EX Series, MX Series, and QFX Series devices with ELS installed. ELS Translator is hosted on the Juniper Networks Customer Support website for EX Series switches, MX Series routers, and QFX Series switches and is available to registered users, internal users, partners, and premium service contract customers. You need to log in using your Juniper Networks username and password to access the ELS Translator.

Click here to access the ELS translator.

If you are upgrading from a version of Junos OS that does not support ELS to a version of Junos OS that supports ELS, we recommend that you update your configuration with the ELS Translator using the following procedure:

1. Log in to your device by using the console port.

   NOTE: Perform this procedure only from the console port. You will lose connectivity to your device if you perform this procedure from a management port or any other interface.

2. Copy the entire existing configuration to another file. Save the file in a remote location. See *Saving a Configuration to a File*.

3. Retain the portion of your existing configuration related to management network connectivity (such as **[edit system]** hierarchy level). Delete all other top-level configuration hierarchy levels (such as the **[edit interfaces]**, **[edit protocols]**, and **[edit vlans]**). Issue the **commit** command to remove the deleted configuration hierarchy levels.

4. Perform the software upgrade. Reboot your device to complete the upgrade. See *Software Installation and Upgrade Overview*.

   NOTE: Ensure that the console port connection is up during the reboot.

5. Click here to access the ELS Translator in a web browser. Follow the instructions on the page to update your configuration.

6. Return to your console port connection. When the switch has rebooted to complete the software upgrade, copy the configuration from the ELS Translator to your switch. See *Uploading a Configuration File*.

7. Commit the new configuration.

NOTE: It is possible that scripts do not translate correctly. Therefore, review translated scripts carefully before loading the converted configuration on your switch or other device.

Layer 2 Next Generation Mode for ACX Series

The Layer 2 Next Generation mode, also called Enhanced Layer 2 Software (ELS), is supported on ACX5048, ACX5096, and ACX5448 routers for configuring Layer 2 features.

The Layer 2 CLI configurations and show commands for ACX5048, ACX5096, and ACX5448 routers differ from those for other ACX Series routers (ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, and ACX4000) and MX Series routers.

For more information about Enhanced Layer 2 Software, see Getting Started with Enhanced Layer 2 Software.

Table 18 on page 20 shows the differences in CLI hierarchy for configuring Layer 2 features in Layer 2 next generation mode.

Table 18: Differences in CLI Hierarchy for Layer 2 Features in Layer 2 Next Generation Mode

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
Bridge Domain	[edit bridge-domains *bridge-domain-name*]	[edit vlans *vlan-name*]
Family **bridge**	[edit interfaces *interface-name* unit *unit-number* family bridge]	[edit interfaces *interface-name* unit *unit-number* family ethernet-switching]
Layer 2 options	[edit bridge-domains *bridge-domain-name* bridge-options]	[edit vlans *vlan-name* switch-options]
Ethernet options	[edit interfaces *interface-name* gigether-options]	[edit interfaces *interface-name* ether-options]
Integrated routing and bridging (IRB)	[edit bridge-domains *bridge-domain-name*] routing-interface irb.*unit*;	[edit vlans *vlan-name*] l3-interface irb.*unit*;
Storm control	[edit vlans *vlan-name* forwarding-options flood filter *filter-name*]	[edit forwarding-options storm-control-profiles] [edit interfaces *interface-name* ether-options] storm-control *name*; recovery-timeout *interval*;
Internet Group Management Protocol (IGMP) snooping	[edit bridge-domains *bridge-domain-name* protocols igmp-snooping]	[edit protocols igmp-snooping vlan *vlan-name*]
Family **bridge** firewall filter	[edit firewall family bridge]	[edit firewall family ethernet-switching]

Table 19 on page 20 shows the differences in **show** commands for Layer 2 features in Layer 2 next generation mode.

Table 19: Differences in show Commands for Layer 2 Features in Layer 2 Next Generation Mode

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
VLAN	**show bridge-domain**	**show vlans**
MAC table	**show bridge mac-table**	**show ethernet-switching table**

Table 19: Differences in show Commands for Layer 2 Features in Layer 2 Next Generation Mode (continued)

Feature	ACX1000, ACX1100, ACX2000, ACX2100, ACX2200, ACX4000, and MX Series Routers	ACX5048, ACX5096, and ACX5448 Routers
MAC table options	**show bridge mac-table** (MAC address, bridge-domain name, interface, VLAN ID, and instance)	**show ethernet-switching table**
Switch port listing with VLAN assignments	**show l2-learning interface**	**show ethernet-switching interfaces**
Kernel state of flush database	**show route forwarding-table family bridge**	**show route forwarding-table family ethernet-switching**

Related Documentation

- *Storm Control on ACX Series Routers Overview*
- *Layer 2 Bridge Domains on ACX Series Overview*
- *Guidelines for Configuring Firewall Filters*
- *IGMP Snooping and Bridge Domains*
- *Understanding Ethernet Link Aggregation on ACX Series Routers*

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

  NOTE: **Link aggregation is not supported on NFX150 devices.**

- Storm control on the physical port for unicast, multicast, and broadcast

  NOTE: **Storm control is not supported on NFX150 devices.**

- STP support, including 802.1d, RSTP, MSTP, and Root Guard

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84

Understanding Layer 2 Broadcasting on Switches

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 10.0.0.0, the broadcast network address is 10.255.255.255. In this case, only devices that belong to the 10.0.0.0 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 21
- *Understanding Storm Control*
- Understanding Bridging and VLANs on Switches on page 84

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 21
- Understanding Bridging and VLANs on Switches on page 84

Ethernet Switching and Layer 2 Transparent Mode Overview

Layer 2 transparent mode provides the ability to deploy the firewall without making changes to the existing routing infrastructure. The firewall is deployed as a Layer 2 switch with multiple VLAN segments and provides security services within VLAN segments. Secure wire is a special version of Layer 2 transparent mode that allows bump-in-wire deployment.

Ethernet switching forwards the Ethernet frames within or across the LAN segment (or VLAN) using the Ethernet MAC address information. Ethernet switching on the SRX1500 device is performed in the hardware using ASICs.

Starting in Junos OS Release 15.1X49-D40, use the **set protocols l2-learning global-mode(transparent-bridge | switching)** command to switch between the Layer 2 transparent bridge mode and Ethernet switching mode. After switching the mode, you must reboot the device for the configuration to take effect.

NOTE: On SRX1500, the default Layer 2 global mode is transparent-bridge mode.

NOTE: Starting in Junos OS Release 15.1X49-D50 and Junos OS Release 17.3R1, the factory-default configuration of the SRX300, SRX320, SRX340, and SRX345 devices is switching mode. When these devices are loaded or reset with the factory-default configuration, these devices are in switching mode.

Starting from 15.1X49-D50 until 15.1X49-D90, when you load or reset the factory-default configuration on SRX300, SRX320, SRX340, and SRX345 devices, these devices are in switching mode. When you delete the Layer 2 global mode configuration on these devices, these devices are in transparent-bridge mode.

Starting with Junos OS Release 15.1X49-D100, when you load or reset the factory-default configuration on SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M devices, these devices are in switching mode. When you delete the Layer 2 global mode configuration on these devices, these devices are in switching mode. To configure the devices to transparent-bridge mode, you must configure the Layer 2 global mode configuration to transparent-bridge mode. Use the command **set protocols l2-learning global-mode transparent-bridge** before rebooting the devices with Junos OS 15.1X49-D100 image.

The Layer 2 protocol supported in switching mode is Link Aggregation Control Protocol (LACP).

You can configure Layer 2 transparent mode on a redundant Ethernet interface. Use the following commands to define a redundant Ethernet interface:

- **set interfaces *interface-name* ether-options redundant-parent *reth-interface-name***
- **set interfaces *reth-interface-name* redundant-ether-options redundancy-group *number***

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40, use the **set protocols l2-learning global-mode(transparent-bridge | switching)** command to switch between the Layer 2 transparent bridge mode and Ethernet switching mode.
15.1X49-D100	Starting with Junos OS Release 15.1X49-D100, when you load or reset the factory-default configuration on SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M devices, these devices are in switching mode. When you delete the Layer 2 global mode configuration on these devices, these devices are in switching mode.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- global-mode (Protocols) on page 975
- l2-learning on page 1012

CHAPTER 2

Configuring Layer 2 Forwarding Tables

Layer 2 Learning and Forwarding for VLANs Overview

Understanding Layer 2 Forwarding Tables on Switches, Routers and NFX Series Devices

You can configure Layer 2 MAC address and VLAN learning and forwarding properties in support of Layer 2 bridging. Unicast media access control (MAC) addresses are learned to avoid flooding the packets to all the ports in a VLAN. A source MAC entry is created in its source and destination MAC tables for each MAC address learned from packets received on ports that belong to the VLAN.

When you configure a VLAN, Layer 2 address learning is enabled by default. The VLAN learns unicast media access control (MAC) addresses to avoid flooding the packets to all the ports in the VLAN. Each VLAN creates a source MAC entry in its source and destination MAC tables for each source MAC address learned from packets received on the ports that belong to the VLAN.

NOTE: **Traffic is not flooded back onto the interface on which it was received. However, because this "split horizon" occurs at a late stage, the packet statistics displayed by commands such as show interfaces queue will include flood traffic.**

You can optionally disable MAC learning either for the entire device or for a specific VLAN or logical interface. You can also configure the following Layer 2 learning and forwarding properties:

- Timeout interval for MAC entries
- Static MAC entries for logical interfaces only
- Limit to the number of MAC addresses learned from a specific logical interface or from all the logical interfaces in a VLAN
- Size of the MAC address table for the VLAN
- MAC accounting for a VLAN

For more information about how to configure VLANs and virtual switches, see "Configuring a VLAN" on page 202 and "Configuring a Layer 2 Virtual Switch on an EX Series Switch" on page 95.

Understanding Layer 2 Forwarding Tables on Security Devices

The SRX Series device maintains forwarding tables that contain MAC addresses and associated interfaces for each Layer 2 VLAN. When a packet arrives with a new source MAC address in its frame header, the device adds the MAC address to its forwarding table and tracks the interface at which the packet arrived. The table also contains the corresponding interface through which the device can forward traffic for a particular MAC address.

If the destination MAC address of a packet is unknown to the device (that is, the destination MAC address in the packet does not have an entry in the forwarding table), the device duplicates the packet and floods it on all interfaces in the VLAN other than the interface on which the packet arrived. This is known as *packet flooding* and is the default behavior for the device to determine the outgoing interface for an unknown destination MAC address. Packet flooding is performed at two levels: packets are flooded to different zones as permitted by configured Layer 2 security policies, and packets are also flooded to different interfaces with the same VLAN identifier within the same zone. The device learns the forwarding interface for the MAC address when a reply with that MAC address arrives at one of its interfaces.

You can specify that the SRX Series device use ARP queries and traceroute requests (which are ICMP echo requests with the time-to-live values set to 1) instead of packet flooding to locate an unknown destination MAC address. This method is considered more secure than packet flooding because the device floods ARP queries and traceroute packets—not the initial packet—on all interfaces. When ARP or traceroute flooding is used, the original packet is dropped. The device broadcasts an ARP or ICMP query to all other devices on the same subnetwork, requesting the device at the specified destination IP address to send back a reply. Only the device with the specified IP address replies, which provides the requestor with the MAC address of the responder.

ARP allows the device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnetwork as the ingress IP address. (The ingress IP address refers to the IP address of the last device to send the packet to the device. The device might be the source that sent the packet or a router forwarding the packet.) Traceroute allows the device to discover the destination MAC address even if the destination IP address belongs to a device in a subnetwork beyond that of the ingress IP address.

When you enable ARP queries to locate an unknown destination MAC address, traceroute requests are also enabled. You can also optionally specify that traceroute requests not be used; however, the device can then discover destination MAC addresses for unicast packets only if the destination IP address is in the same subnetwork as the ingress IP address.

Whether you enable ARP queries and traceroute requests or ARP-only queries to locate unknown destination MAC addresses, the SRX Series device performs the following series of actions:

1. The device notes the destination MAC address in the initial packet. The device adds the source MAC address and its corresponding interface to its forwarding table, if they are not already there.

2. The device drops the initial packet.

3. The device generates an ARP query packet and optionally a traceroute packet and floods those packets out all interfaces except the interface on which the initial packet arrived.

   ARP packets are sent out with the following field values:

   - Source IP address set to the IP address of the IRB
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the MAC address of the IRB
   - Destination MAC address set to the broadcast MAC address (all **0xf**)

   Traceroute (ICMP echo request or ping) packets are sent out with the following field values:

   - Source IP address set to the IP address of the original packet
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the source MAC address of the original packet
   - Destination MAC address set to the destination MAC address of the original packet
   - Time-to-live (TTL) set to **1**

4. Combining the destination MAC address from the initial packet with the interface leading to that MAC address, the device adds a new entry to its forwarding table.

5. The device forwards all subsequent packets it receives for the destination MAC address out the correct interface to the destination.

Related Documentation

- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- Layer 2 Transparent Mode Overview on page 377
- Understanding Integrated Routing and Bridging on page 445

- Example: Configuring an IRB Interface on a Security Device on page 452
- Example: Configuring the Default Learning for Unknown MAC Addresses on page 57

Configuring Forwarding Mode on Switches

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 902

Understanding the Unified Forwarding Table on QFX Switches

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address. The Unified Forwarding Table (UFT) feature enables you to allocate forwarding table resources to optimize the memory available for different address types based on the needs of your network. You can choose to allocate a higher percentage of memory for one type of address or another.

- Using the Unified Forwarding Table to Optimize Address Storage on page 30
- Understanding the Allocation of MAC Addresses and Host Addresses on page 31
- Understanding Ternary Content Addressable Memory (TCAM) and Longest Prefix Match Entries on page 36
- Host Table Example for Profile with Heavy Layer 2 Traffic on page 36

Using the Unified Forwarding Table to Optimize Address Storage

On the QFX5100, EX4600, EX4650, QFX5110, QFX5200, and QFX5120 switches, you can control the allocation of forwarding table memory available to store the following:

- MAC addresses—In a Layer 2 environment, the switch learns new MAC addresses and stores them in a MAC address table
- Layer 3 host entries—In a Layer 2 and Layer 3 environment, the switch learns which IP addresses are mapped to which MAC addresses; these key-value pairs are stored in the Layer 3 host table.
- Longest prefix match (LPM) table entries—In a Layer 3 environment, the switch has a routing table and the most specific route has an entry in the forwarding table to associate a prefix or netmask to a next hop. Note, however, that all IPv4 /32 prefixes and IPv6 /128 prefixes are stored in the Layer 3 host table.

UFT essentially combines the three distinct forwarding tables to create one table with flexible resource allocation. You can select one of five forwarding table profiles that best

meets your network needs. Each profile is configured with different maximum values for each type of address. For example, for a switch that handles a great deal of Layer 2 traffic, such as a virtualized network with many servers and virtualized machines, you would likely choose a profile that allocates a higher percentage of memory to MAC addresses. For a switch that operates in the core of a network, participates in an IP fabric, you probably want to maximize the number of routing table entries it can store. In this case, you would choose a profile that allocates a higher percentage of memory to longest match prefixes. The QFX5200 switch supports a custom profile that allows you to partition the four available shared memory banks with a total of 128,000 entries among MAC addresses, Layer 3 host addresses, and LPM prefixes.

NOTE: Support for QFX5200 switches was introduced in Junos OS Release 15.1x53-D30. The QFX5200 switch is not supported on Junos OS Release 16.1R1.

Understanding the Allocation of MAC Addresses and Host Addresses

All five profiles are supported, each of which allocates different amounts of memory for Layer 2 or Layer 3 entries, enabling you choose one that best suits the needs of your network. The QFX5200 and QFX5210 switches, however, supports different maximum values for each profile from the other switches. For more information about the custom profile, see "Configuring the Unified Forwarding Table on Switches" on page 37.

NOTE: The default profile is **l2-profile-three**, which allocates equal space for MAC Addresses and Layer 3 host addresses. On QFX5100, EX4600, QFX5110, and QFX5200 switches, the space is equal to 16,000 IPv4 entries for the LPM table, and on QFX5210 switches, the space is equal to 32,000 IPv4 entries for the LPM table. For the **lpm-profile** the LPM table size is equal to 256,000 IPv4 entries.

NOTE: Starting with Junos OS Release 18.1R1 on the QFX5210-64C switch, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.

NOTE: Starting with Junos OS Release 18.3R1 on the QFX5120 and EX4650 switches, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.

NOTE: On QFX5100, EX4600, EX4650, QFX5110, QFX5200, QFX5120, and QFX5210-64C switches, IPv4 and IPv6 host routes with ECMP next hops are stored in the host table.

BEST PRACTICE: If the host or LPM table stores the maximum number of entries for any given type of entry, the entire shared table is full and is unable to accommodate *any* entries of any other type. 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 20 on page 32 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5100 and EX4600 switches.

Table 20: Unified Forwarding Table Profiles on QFX5100 and EX4600 Switches

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
lpm-profilewith **unicast-in-lpm** option	32K	(stored in LPM table)	(stored in LPM table)	8K	8K	4K	4K

Table 21 on page 32 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5110 switches.

Table 21: Unified Forwarding Table Profiles on QFX5110 Switches

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

Table 22 on page 33 lists the LPM table size variations for the QFX5110 switch depending on the prefix entries.

Table 22: LPM Table Size Variations on QFX5110 Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 LPM<= /32	IPv6 LPM <= /64	IPv6 LPM > /64
0	16K	8K	0K
1	12K	6K	1K
2	8K	4K	2K
3	4K	2K	3K
4	0K	0K	4K

Table 23 on page 33 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5200-32C switches.

Table 23: Unified Forwarding Table Profiles on QFX5200-32C Switches

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)	Exact-Match
l2-profile-one	136K	8K	4K	4K	4K	2K	2K	0
l2-profile-two	104K	40K	20K	20K	20K	10K	10K	0
l2-profile-three (default)	72K	72K	36K	36K	36K	18K	18K	0
l3-profile	40K	104K	52K	52K	52K	26K	26K	0
lpm-profile	8K	8K	4K	4K	4K	2K	2K	0

Table 24 on page 33 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5200-48Y switches.

Table 24: Unified Forwarding Table Profiles on QFX5200-48Y Switches

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	136K	8K	4K	4K	4K	2K	2K
l2-profile-two	104K	40K	20K	20K	20K	10K	10K

Table 24: Unified Forwarding Table Profiles on QFX5200-48Y Switches (continued)

Profile Name	MAC Table	Host Table (unicast and multicast addresses)					
l2-profile-three (default)	72K	72K	36K	36K	36K	18K	18K
l3-profile	40K	104K	52K	52K	52K	26K	26K
lpm-profile	8K	8K	4K	4K	4K	2K	2K

Table 25 on page 34 lists the LPM table size variations for the QFX5200-48Y switch depending on the prefix entries.

Table 25: LPM Table Size Variations on QFX5200-48Y Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 LPM<= /32	IPv6 LPM <= /64	IPv6 LPM > /64
0	16K	8K	0K
1	12K	6K	1K
2	8K	4K	2K
3	40K	2K	3K
4	0K	0K	4K

Table 26 on page 34 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5210-64C switches.

Table 26: Unified Forwarding Table Profiles on QFX5210-64C Switches

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)	Exact Match
l2-profile-one	264K	8K	4K	4K	4K	2K	2K	0K
l2-profile-two	200K	72K	36K	36K	36K	18K	18K	0K
l2-profile-three (default)	136K	136K	72K	72K	72K	36K	36K	0K
l3-profile	72K	200K	100K	100K	100K	50K	50K	0K

Table 27 on page 35 lists the profiles you can choose and the associated maximum values for the MAC address and host table entries on QFX5120 and EX4650 switches.

Table 27: Unified Forwarding Table Profiles on QFX5120 and EX4650 Switches

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

Table 28 on page 35 lists the LPM table size variations for the QFX5210-64C switch depending on the prefix entries.

Table 28: LPM Table Size Variations on QFX5210-64C Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 LPM<= /32	IPv6 LPM <= /64	IPv6 LPM > /64
0	32K	16K	0K
1	28K	14K	1K
2	24K	12K	2K
3	20K	10K	3K
4	0K	0K	4K

Table 29 on page 35 lists the Layer 3 Defip table size variations for the QFX5120 and EX4650 switches depending on the changing IPv6/128 prefix entries.

Table 29: LPM Table Size Variations on QFX5210-64C and EX4650 Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 LPM<= /32	IPv6 LPM <= /64	IPv6 LPM > /64
0	32K	16K	0K
2	24K	12K	2K
4	16K	8K	4K
6	8K	4K	6K
8	0K	0K	8K

Understanding Ternary Content Addressable Memory (TCAM) and Longest Prefix Match Entries

You can further customize non-LPM profiles by configuring the space available for ternary content addressable memory (TCAM) to allocate more memory for longest prefix match entries. You can change the number of entries allocated to these IPv6 addresses, essentially allocating more or less space for LPM IPv4 entries with any prefix length or IPv6 entries with prefix lengths of 64 of shorter. For more information about how to change the default parameters of the TCAM memory space for LPM entries, see "Configuring the Unified Forwarding Table on Switches" on page 37.

NOTE: The option to adjust TCAM space is not supported on the longest prefix match (LPM) or custom profiles. However, for the LPM profile, you can configure TCAM space not to allocate any memory for IPv6 entries with prefix lengths of 65 or longer, thereby allocating that memory space only for IPv4 routes or IP routes with prefix lengths equal to or less than 64 or a combination of the two types of prefixes.

NOTE: Starting with Junos OS Release 18.1R1 on QFX5210 switches, you can configure TCAM space to allocate a maximum of 8,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 2,000 entries. Starting with Junos OS Release 13.2X51-D15, you can configure TCAM space to allocate a maximum of 4,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 1,000 entries. Previous to Junos OS Release 13.2X51-D15, you could allocate only a maximum of 2,048 entries for IPv6 the IPv6 prefixes with lengths in the range /65 to /127 range. The default value was 16 entries for these types of IPv6 prefixes.

On Junos OS Releases 13.2x51-D10 and 13.2x52D10, the procedure to change the default value of 16 entries differs from later releases, where the maximum and default values are higher. For more information about that procedure, see "Configuring the Unified Forwarding Table on Switches" on page 37

Host Table Example for Profile with Heavy Layer 2 Traffic

Table 30 on page 36 lists various valid combinations that the host table can store if you use the **l2-profile-one** profile on QFX5100 and EX4600 switches. This profile allocates the percentage of memory to Layer 2 addresses. Note that the default values might be different on other switches. Each row in the table represents a case in which the host table is full and cannot accommodate any more entries.

Table 30: Example Host Table Combinations Using l2-profile-one on QFX5100 and EX4600 Switches

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

Table 30: Example Host Table Combinations Using l2-profile-one on QFX5100 and EX4600 Switches (continued)

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

Release History Table

Release	Description
18.1R1	Starting with Junos OS Release 18.1R1 on the QFX5210-64C switch, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.
18.1R1	Starting with Junos OS Release 18.3R1 on the QFX5120 and EX4650 switches, for all these profiles, except for the **lpm-profile** the longest prefix match (LPM) table size is equal to 32,000 IPv4 entries.
18.1R1	Starting with Junos OS Release 18.1R1 on QFX5210 switches, you can configure TCAM space to allocate a maximum of 8,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 2,000 entries.
13.2X51-D15	Starting with Junos OS Release 13.2X51-D15, you can configure TCAM space to allocate a maximum of 4,000 IPv6 entries with prefix lengths of 65 or longer. The default value is 1,000 entries.

Related Documentation

- Configuring the Unified Forwarding Table on Switches on page 37

Configuring the Unified Forwarding Table on Switches

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address stored in the tables. The Unified Forwarding Table feature lets you optimize how your switch allocates forwarding-table memory for different types of addresses. You can choose one of five unified forwarding table profiles. Each profile allocates a different maximum amount of memory for Layer 2, Layer 3 host, and longest prefix match (LPM) entries. In addition to selecting a profile, you can also select how much additional memory to allocate for LPM entries.

Two profiles allocate higher percentages of memory to Layer 2 addresses. A third profile allocates a higher percentage of memory to Layer 3 host address, while a fourth profile allocates a higher percentage of memory to LPM entries. There is a default profile configured that allocates an equal amount of memory to Layer 2 and Layer 3 host addresses with the remainder allocated to LPM entries. For a switch in a virtualized network that handles a great deal of Layer 2 traffic, you would choose a profile that

allocates a higher percentage of memory to Layer 2 addresses. For a switch that operates in the core of the network, you would choose a profile that allocates a higher percentage of memory to LPM entries.

On QFX5200 and QFX5210-64C switches only, you can also configure a custom profile that allows you to partition shared memory banks among the different types of forwarding table entries. On QFX5200 switches, these shared memory banks have a total memory equal to 128,000 IPv4 unicast addresses. On QFX5210 switches, these shared memory banks have a total memory equal to 256,000 IPv4 unicast addresses. For more information about configuring the custom profile, see "Example: Configuring a Unified Forwarding Table Custom Profile on QFX Series Switches" on page 47.

Configuring a Unified Forwarding Table Profile

To configure a unified forwarding table profile:

Specify a forwarding-table profile.

```
[edit chassis forwarding-options]
user@switch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@switch# set l2-profile-one
```

CAUTION: When you configure and commit a profile, in most cases the Packet Forwarding Engine automatically restarts and all the data interfaces on the switch go down and come back up (the management interfaces are unaffected).

Starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, for a Virtual Chassis or Virtual Chassis Fabric (VCF) comprised of EX4600 or QFX5100 switches, the Packet Forwarding Engine in member switches does not automatically restart upon configuring and committing a unified forwarding table profile change. This behavior avoids Virtual Chassis or VCF instability after the change propagates to member switches and multiple Packet Forwarding Engines automatically restart at the same time. Instead, a message is displayed at the CLI prompt and logged to the switch's system log to notify you that the profile change does not take effect until the next time you reboot the Virtual Chassis or VCF. We recommend that you plan to make profile changes only when you can perform a Virtual Chassis or VCF system reboot immediately after committing the configuration update. Otherwise, the Virtual Chassis or VCF could become inconsistent if one or more members have a problem and restart with the new configuration before a planned system reboot activates the change on all members.

NOTE: You can configure only one profile for the entire switch.

NOTE: The l2-profile-three is configured by default.

NOTE: 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. Keep in mind that 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..

Configuring the Memory Allocation for Longest Prefix Match Entries

In addition to choosing a profile, you can further optimize memory allocation for longest prefix match (LPM) entries by configuring how many IPv6 prefixes to store with lengths from /65 through /127. The switch uses LPM entries during address lookup to match addresses to the most-specific (longest) applicable prefix. Prefixes of this type are stored in the space for ternary content addressable memory (TCAM). Changing the default parameters makes this space available for LPM entries. Increasing the amount of memory available for these IPv6 prefixes reduces by the same amount how much memory is available to store IPv4 unicast prefixes and IPv6 prefixes with lengths equal to or less than 64.

The procedures for configuring the LPM table are different, depending on which version of Junos OS you are using. In the initial releases that UFT is supported, Junos OS Releases 13.2X51-D10 and 13.2X52-10, you can only increase the amount of memory allocated to IPv6 prefixes with lengths from /65 through /127 for any profile, except for **lpm-profile**. Starting with Junos OS Release 13.2X51-D15, you can also allocate either less or no memory for IPv6 prefixes with lengths in the range /65 through /127, depending on which profile is configured. For the **lpm-profie**, however, the only change you can make to the default parameters is to allocate no memory for these types of prefixes.

- Configuring the LPM Table With Junos OS Releases 13.2X51-D10 and 13.2X52-D10 on page 39
- Configuring the LPM Table With Junos OS Release 13.2x51-D15 and Later on page 40

Configuring the LPM Table With Junos OS Releases 13.2X51-D10 and 13.2X52-D10

In Junos OS Releases 13.2x51-D10 and 13.2X52-D10, by default, the switch allocates memory for 16 IPv6 with prefixes with lengths in the range /65 through /127. You can configure the switch to allocate more memory for IPv6 prefixes with lengths in the range /65 through /127.

To allocate more memory for IPv6 prefixes in the range /65 through /127:

1. Choose a forwarding table profile.

```
[edit chassis forwarding-options]
user@swtitch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@swtitch# set l2-profile-one
```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

```
[edit chassis forwarding-options profile-name]
user@swtitch# set num-65-127-prefix number
```

For example, to specify to allocate memory for 32 IPv6 prefixes in the range /65 through 127:

```
[edit chassis forwarding-options l2-profile-one]
user@switch# set num-65-127-prefix 2
```

NOTE: When you configure and commit the **num-65-127-prefix** *number* statement, all the data interfaces on the switch restart. The management interfaces are unaffected.

The **num-65-127-prefix** *number* statement is not supported on the **lpm-profile**.

Configuring the LPM Table With Junos OS Release 13.2x51-D15 and Later

Configuring Layer 2 and Layer 3 Profiles With Junos OS Release 13.2x51-D15 or Later

Starting in Junos OS Release 13.2X51-D15, you can configure the switch to allocate forwarding table memory for as many as 4,000 IPv6 prefixes with lengths in the range /65 through /127 for any profile other than the **lpm-profile** or **custom-profile**. You can also specify to allocate no memory for these IPv6 entries. The default is 1,000 entries for IPv6 prefixes with lengths in the range /65 through /127. Previously, the maximum you could configure was for 2,048 entries for IPv6 prefixes with lengths in the range /65 through /127. The minimum number of entries was previously 16, which was the default.

To specify how much forwarding table memory to allocate for IPv6 prefixes with length in the range /65 through /127:

1. Choose a forwarding table profile.

```
[edit chassis forwarding-options]
user@swtitch# set profile-name
```

For example, to specify the profile that allocates the highest percentage of memory to Layer 2 traffic:

```
[edit chassis forwarding-options]
user@swtitch# set l2-profile-one
```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

```
[edit chassis forwarding-options profile-name]
user@swtitch# set num-65-127-prefix number
```

For example, to specify to allocate memory for 2,000 IPv6 prefixes in the range /65 through 127:

```
[edit chassis forwarding-options l2-profile-one]
user@switch# set num-65-127-prefix 2
```

Starting with Junos OS Release 13.2X51-D15, you can use the **num-65-127-prefix** statement to allocate entries. Table 31 on page 41 shows the numbers of entries that you can allocate. Each row represents a case in which the table is full and cannot accommodate any more entries.

Table 31: LPM Table Combinations for L2 and L3 profiles With Junos OS 13.2X51-D15 and Later

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

CAUTION: When you configure and commit a profile change with the **num-65-127-prefix** *number* statement, the Packet Forwarding Engine automatically restarts and all the data interfaces on the switch go down and come back up (the management interfaces are unaffected).

However, starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, Packet Forwarding Engines on switches in a Virtual Chassis or Virtual Chassis Fabric (VCF) do not automatically restart upon configuring a unified forwarding table profile change. This behavior avoids Virtual Chassis or VCF instability after the change propagates to member switches and multiple Packet Forwarding Engines automatically restart at the same time. Instead, a message is displayed at the CLI prompt and logged to the switch's system log to notify you that the profile change does not take effect until the next time you reboot the Virtual Chassis or VCF. We recommend that you plan to make profile changes only when you can perform a Virtual Chassis or VCF

system reboot immediately after committing the configuration update. Otherwise, the Virtual Chassis or VCF could become inconsistent if one or more members have a problem and restart with the new configuration before a planned system reboot activates the change on all members.

Configuring the lpm-profile With Junos OS Release 13.2x51-D15 and Later

Starting with Junos OS Release 13.2X51-D15 you can configure the **lpm-profile** profile not to allocate any memory for IPv6 entries with prefix lengths from /65 through /127. These are the default maximum values allocated for LPM memory for the **lpm-profile** by address type:

- 128K of IPv4 prefixes
- 16K of IPv6 prefixes (all lengths)

NOTE: The memory allocated for each address type represents the maximum default value for all LPM memory.

To configure the **lpm-profile** not to allocate forwarding-table memory for IPv6 entries with prefixes from /65 through /127, thus allocating more memory for IPv4:

Specify to disable forwarding-table memory for IPv6 prefixes with lengths in the range /65 through /127.

```
[edit chassis forwarding-options lpm-profile]
user@switch# set prefix-65-127-disable
```

For example, on the QFX5100 and EX4600 switches only, if you use the **prefix-65-127-disable** option, each of the following combinations are valid:

- 100K IPv4 and 28K IPv6 /64 or shorter prefixes.
- 64K IPv4 and 64K IPv6 /64 or shorter prefixes.
- 128K IPv4 and 0K IPv6 /64 or shorter prefixes.
- 0K IPv4 and 128K IPv6 /64 or shorter prefixes.

NOTE: On the QFX5200 switches, when you configure the **prefix-65-127-disable** statement, the maximum number of IPv6 entries with prefixes equal to or shorter than 64 is 98,000.

Configuring the lpm-profile With Junos OS Release 14.1x53-D30 and Later

Starting in Junos OS Release 15.1X53-D30, you can configure the **lpm-profile** profile to store unicast IPv4 and IPv6 host addresses in the LPM table , thereby freeing memory in the host table. Unicast IPv4 and IPv6 addresses are stored in the LPM table instead of the host table, as shown in Table 32 on page 43 for QFX5100 and EX4600 switches.

(Platform support depends on the Junos OS release in your installation.) You can use this option in conjunction with the option to allocate no memory in the LPM table for IPv6 entries with prefix lengths in the range /65 through /127. Together, these options maximize the amount of memory available for IPv4 unicast entries and IPv6 entries with prefix lengths equal to or less than 64.

Table 32: lpm-profile with unicast-in-lpm Option for QFX5100 and EX4600 Switches

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)
No	32K	0	0	8K	8K	4K	4K	128K	16K	16K
Yes	32K	0	0	8K	8K	4K	4K	128K	128K	0

Starting with Junos Release 18.1R1, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**.

Table 33 on page 43 lists the situations in which the **prefix-65-127-disable** statement should be enabled or disabled.

Table 33: LPM Table Size Variations on QFX5200-48Y Switches

Profile Name	Prefix Entries		
num-65-127-prefix	IPv4 <= /32	IPv6 <= /64	IPv6 > /64
Enabled	> 128K (minimum guaranteed)	98K	0K
Disabled	128K	16K	16K

On QFX5120 and EX4600 switches, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**

Table 34 on page 43 lists the situations in which the **prefix-65-127-disable** statement should be enabled or disabled.

Table 34: LPM Table Size Variations on QFX5120 and EX4650 Switches

Profile Name	Prefix Entries		
prefix-65-127-disable	IPv4 <= /32	IPv6 <= /64	IPv6 > /64
Enabled	351K (360,000 approximate)	168K (172,000 approximate)	0K
Disabled	168K (172,000 approximate)	64K (65,524 approximate)	64K (65,524 approximate)

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 stored, the entire LPM table is full and no additional addresses (IPv4 or IPv6) can be stored.

To configure the **lpm-profile** to store unicast IPv4 entries and IPv6 entries with prefix lengths equal to or less than 64 in the LPM table:

1. Specify the option to store these entries in the LPM table.

   ```
   [edit chassis forwarding-options lpm-profile]
   user@switch# set unicast-in-lpm
   ```

2. (Optional) Specify to allocate no memory for in the LPM table for IPv6 prefixes with length in the range /65 through /127:

   ```
   [edit chassis forwarding-options lpm-profile]
   user@switch# set prefix-65-127-disable
   ```

Configuring Non-LPM Profiles on QFX5120 and EX4650 Switches

For non-LPM profiles, each profile provides the option of reserving a portion of the 16K L3-defip table to store IPv6 Prefixes > 64. Because these are 128-bit prefixes, you can have maximum of 8k IPv6/128 entries in the l3-defip table.

1. Choose a forwarding table profile.

   ```
   [edit chassis forwarding-options]
   user@swtitch# set profile-name
   ```

   For example, to specify the profile that allocates the highest percentage of memory to Layer 3 traffic:

   ```
   [edit chassis forwarding-options]
   user@swtitch# set l3-profile
   ```

2. Select how much memory to allocate for IPv6 prefixes in the range /65 thorugh 127.

   ```
   [edit chassis forwarding-options profile-name]
   user@swtitch# set num-65-127-prefix number
   ```

   For example, to specify to allocate memory for 2,000 IPv6 prefixes in the range /65 through 127:

   You can choose between 0 and 4, 1 being the default.

   ```
   [edit chassis forwarding-options l3-profile]
   user@switch# set num-65-127-prefix 1
   ```

Release History Table

Release	Description
18.1R1	Starting with Junos Release 18.1R1, you cannot set configure a prefix for the **num-65-127-prefix** statement on non-LPM profiles. You can only enable or disable the **prefix-65-127-disable** statement for the **lpm-profile**.
14.1X53-D40	Starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, for a Virtual Chassis or Virtual Chassis Fabric (VCF) comprised of EX4600 or QFX5100 switches, the Packet Forwarding Engine in member switches does not automatically restart upon configuring and committing a unified forwarding table profile change.
13.2X51-D15	Starting with Junos OS Release 13.2X51-D15, you can also allocate either less or no memory for IPv6 prefixes with lengths in the range /65 through /127, depending on which profile is configured.
13.2X51-D15	Starting in Junos OS Release 13.2X51-D15, you can configure the switch to allocate forwarding table memory for as many as 4,000 IPv6 prefixes with lengths in the range /65 through /127 for any profile other than the **lpm-profile** or **custom-profile**.
13.2X51-D15	Starting with Junos OS Release 13.2X51-D15, you can use the **num-65-127-prefix** statement to allocate entries.

Related Documentation

- *Understanding the Unified Forwarding Table*

Configuring Double-Tagged VLANs on Layer 3 Logical Interfaces

Junos OS supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same switch but preventing them from being in the same routing or bridging domain. 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 network nodes receiving the frames can detect which VLAN the frames belong to.

You can configure double VLAN tags (that is, an inner and an outer tag) on a Layer 3 logical interface (sometimes called a "Layer 3 subinterface").

Support for double-tagging VLANs on Layer 3 logical interfaces includes:

- Configuration of an IPv4, an IPv6, or an **mpls** family on the logical interface
- Configuration over an aggregated Ethernet interface
- Configuration of multiple logical interfaces on a single physical interface

NOTE: This feature does not include support for the following:

- VLAN rewrite (**input-vlan-map** or **output-vlan-map**)
- TPID configuration (on physical or logical interfaces)
- **native-inner-vlan-id**; **outer-vlan-id-list**; **inner-vlan-id-list**; or **vlan-id-range**

To configure a double-tagged Layer 3 logical interface:

1. Apply flexible VLAN tagging to the physical interface:

```
[edit]
user@switch# set interfaces interface-name flexible-vlan-tagging
```

2. Configure inner and outer VLAN tags on the logical interface:

```
[edit]
user@switch# set interfaces interface-name unit logical-unit-number vlan-tags outer vlan-id
user@switch# set interfaces interface-name unit logical-unit-number vlan-tags inner vlan-id
```

3. Set the family type and, if needed, the address on the logical interface:

```
[edit]
user@switch# set interfaces interface-name unit logical-unit-number family family-type
address address
```

Related Documentation

- Configuring VLANs on Switches with Enhanced Layer 2 Support on page 97

Example: Configuring a Unified Forwarding Table Custom Profile on QFX Series Switches

Traditionally, forwarding tables have been statically defined and have supported only a fixed number of entries for each type of address. The Unified Forwarding Table (UFT) feature enables you to optimize how forwarding-table memory is allocated to best suit the needs of your network. This example shows how to configure a Unified Forwarding Table profile that enables you to partition four shared hash memory banks among three different types of forwarding-table entries: MAC addresses, Layer 3 host addresses, and longest prefix match (LPM).

The UFT feature also supports five profiles that each allocate a specific maximum amount of memory for each type of forwarding table entry. Some profiles allocate more memory to Layer 2 entries, while other profiles allocate more memory to Layer 3 or LPM entries. The maximum values for each type of entry are fixed in these profiles. With the custom profile, you can designate one or more shared memory banks to store a specific type of forwarding-table entry. You can configure as few as one or as many as four memory banks in a custom profile. The custom profile thus provides even more flexibility in enabling you to allocate forwarding-table memory for specific types of entries.

Requirements

This example uses the following hardware and software components:

- One QFX5200 switch
- Junos OS Release 15.1x53-D30 or later.

Before you configure a custom profile, be sure you have:

- Configured interfaces

Overview

The Unified Forwarding Table custom profile enables you to allocate forwarding-table entries among four banks of shared hash tables with a total memory equal to 128,000 unicast IPv4 addresses, or 32,000 entries for each bank. Specifically, you can allocate one or more of these shared banks to store a specific type of forwarding-table entry. The custom profile does not affect the dedicated hash tables. Those tables remain fixed with 8,000 entries allocated to Layer 2 addresses, the equivalent of 8,000 entries allocated to IPv4 addresses, and the equivalent of 16,000 entries allocated to longest prefix match (LPM) addresses.

In this example, you allocate two memory banks to Layer 3 host addresses, and two memory banks to LPM entries. This means that no shared hash table memory is allocated

for Layer 2 addresses. Only the dedicated hash table memory is allocated for Layer 2 addresses in this scenario.

Configuration

To configure a custom profile for the Unified Forwarding Table feature on a QFX5200 switch that allocates two shared memory banks for Layer 3 host address and two shared memory banks for LPM entries, perform these tasks:

- Configuring the Custom Profile on page 48
- Configuring the Allocation of Shared Memory Banks on page 48
- Results on page 49

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode. A commit check is performed to ensure that you have allocated forwarding-table space for no more than four memory banks.

CAUTION: When you configure and commit a profile, the Packet Forwarding Engine restarts and all the data interfaces on the switch go down and come back up.

```
user@switch# set chassis forwarding-options custom-profile
user@switch# set chassis forwarding-options custom-profile l2-entries num-banks 0
user@switch# set chassis forwarding-options custom-profile l3-entries num-banks 2
user@switch# set chassis forwarding-options custom-profile lpm-entries num-banks 2
```

Configuring the Custom Profile

Step-by-Step Procedure

To create the custom profile:

1. Specify the **custom-profile** option.

```
[edit chassis forwarding-options]
user@switch# set custom-profile
```

Configuring the Allocation of Shared Memory Banks

Step-by-Step Procedure

To allocate memory for specific types of entries for the shared memory banks:

1. Specify to allocate no shared bank memory for Layer 2 entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set l2-entries num-banks 0
```

2. Specify to allocate two shared memory banks (or the equivalent of 64,000 IPv4 entries) for Layer 3 host entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set l3-entries num-banks 2
```

3. Specify to allocate two shared memory banks (or the equivalent of 64,000 IPv4 entries) for LPM entries.

```
[edit chassis forwarding-options custom-profile]
user@switch# set lpm-entries numer-banks 2
```

Results

From configuration mode, confirm your configuration by entering the show chassis forwarding-options command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch# show chassis forwarding-profile
custom-profile {
  l2-entries {
    num-banks 0;
  }
  l3-entries {
    num-banks 2;
  }
  lpm-entries {
    num-banks 2
  }
}
```

If you are done configuring the switch, enter **commit** from configuration mode

CAUTION: The Packet Forwarding Engine will restart and all the data interfaces on the switch will go down and come back up.

Verification

Confirm that the configuration is working properly.

- Checking the Parameters of the Custom Profile on page 49

Checking the Parameters of the Custom Profile

Purpose Verify that the custom profile is enabled.

Action

```
user@switch> show chassis forwarding-options
UFT Configuration:
custom-profile
Configured custom scale:
Entry type                 Total scale(K)
L2(mac)                    8
L3 (unicast & multicast)        72
Exact Match                 0
Longest Prefix Match (lpm)        80
num-65-127-prefix = 1K
-------------Bank details for various types of entries------------
Entry type                   Dedicated Bank Size(K)      Shared Bank Size(K)
L2 (mac)                     8                          32 * num shared banks
L3 (unicast & multicast          8                          32 * num shared banks
Exact match                   0                          16 * num shared banks
Longest Prefix match(lpm)           16                        32 * num shared banks
```

Meaning

The output shows that the custom profile is enabled as configured with two shared memory banks designated for Layer 3 host entries; two shared memory banks designated for LPM entries; and no shared memory allocated for Layer 2 entries.

The total scale(K) field shows the total allocation of memory, that is, the amount allocated through the shared memory banks plus the amount allocated through the dedicated hash tables. The amount allocated through the dedicated hash tables is fixed and cannot be changed. Therefore, Layer 2 entries have 8K of memory allocated only through the dedicated hash table. Layer 3 host entries have 64K of memory allocated through two shared memory banks plus 8K through the dedicated hash table, for a total of 72K of memory. LPM entries have 64K of memory allocated through two shared memory banks plus 16K through the dedicated hash table, for a total of 80K of memory.

Related Documentation

- *Understanding the Unified Forwarding Table*

CHAPTER 3

Configuring MAC Addresses

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 21
- Understanding MAC Learning on page 61

Understanding MAC Address Assignment on an EX Series Switch

This topic describes MAC address assignment for interfaces on standalone Juniper Networks EX Series Ethernet Switches. For information regarding MAC address assignments in a Virtual Chassis, see *Understanding MAC Address Assignment on a Virtual Chassis*.

MAC addresses are used to identify network devices at Layer 2. Because all Layer 2 traffic decisions are based on an interface's MAC address, understanding MAC address assignment is important to understanding how network traffic is forwarded and received by the switch. For additional information on how a network uses MAC addresses to forward and receive traffic, see "Understanding Bridging and VLANs on Switches" on page 84.

A MAC address comprises six groups of two hexadecimal digits, with each group separated from the next group by a colon—for instance, aa:bb:cc:dd:ee:00. The first five groups of hexadecimal digits are derived from the switch and are the same for all interfaces on the switch.

The assignment of a unique MAC address to each network interface helps ensure that functions that require MAC address differentiation—such as redundant trunk groups (RTGs), Link Aggregation Control Protocol (LACP), and general monitoring functions—can properly function.

On switches that use line cards, this MAC addressing scheme differentiates the Layer 2 interfaces on different line cards in the switch.

For EX Series switches, the first five groups of hexadecimal digits are determined when the switch is manufactured. The switch then assigns a unique MAC address to each interface by assigning a unique identifier as the last group of hexadecimal digits. The assignment depends on how the interface is configured. The switch uses a different

pattern to distinguish between an interface that is configured as any of a routed VLAN interface (RVI), a virtual management Ethernet (VME) interface, or an aggregated Ethernet interface or is not configured as any of an RVI, a VME, or as an aggregated Ethernet interface.

For aggregated Ethernet interfaces, the MAC address assignment remains constant regardless of whether the configuration of the interface is Layer 2 or Layer 3.

NOTE: In Junos OS Release 11.3 and later releases through Release 12.1, the MAC address assignment for aggregated Ethernet interfaces changes if the interface is changed from Layer 2 to Layer 3 or the reverse. Starting with Junos Release 12.2, the MAC address assignment for aggregated Ethernet interfaces remains constant regardless of whether the interface is Layer 2 or Layer 3.

NOTE: Prior to Junos OS Release 11.3, MAC addresses for Layer 2 interfaces could be shared between interfaces and RVIs on different line cards in the same switch. However, if you upgrade from Junos OS Release 11.2 or earlier to Junos OS Release 11.3 or later on a switch that supports line cards, the MAC addresses of these interfaces will change.

MAC addresses are assigned to interfaces automatically—no user configuration is possible or required. You can view MAC addresses assigned to interfaces using the **show interfaces** command.

Related Documentation

- *Interfaces Overview for Switches*

Configuring the Size of the MAC Address Table

You can modify the size of the MAC address table for each VLAN. The default table size is 5120 addresses. The minimum you can configure is 16 addresses, and the maximum is 1,048,575 addresses.

If the MAC table limit is reached, new addresses can no longer be added to the table. Unused MAC addresses are removed from the MAC address table automatically. This frees space in the table, allowing new entries to be added.

To modify the size of the MAC table, include the **mac-table-size *limit*** statement at the **[edit vlans *vlan-name* switch-options]** hierarchy level:

```
[edit]
vlans {
  vlan-name {
    domain-type bridge;
    switch-options {
      mac-table-size limit {
        packet-action drop;
      }
```

```
    }
  }
}
```

Related Documentation

- *Disabling MAC Learning for a Bridge Domain or Logical Interface*
- *Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain*
- *Limiting MAC Addresses Learned from an Interface in a Bridge Domain*
- *Enabling MAC Accounting for a Bridge Domain*

Configuring MAC Move Parameters

When a MAC address appears on a different physical interface or within a different unit of the same physical interface and this behavior occurs frequently, it is considered a MAC move. You can configure the router to report a MAC address move based on the following parameters: the number of times a MAC address move occurs, a specified period of time over which the MAC address move occurs, and specified number of times a MAC address move occurs in one second. You can only configure the **global-mac-move** statement at the global hierarchy level.

To globally disable the MAC move action feature, include the **disable-action** statement at the **[edit protocols l2-learning global-mac-move]**. This disables the MAC move action feature, while MAC move detection exists.

To configure the time duration after which the port will be unblocked, include the **reopen-time** statement at the **[edit protocols l2-learning global-mac-move]**. The default reopen timer is 180 second.

To configure MAC address move reporting if the MAC address moves at least a specified number of times in one second, include the **threshold-time** statement at the **[edit protocols l2-learning global-mac-move]** hierarchy level. The default threshold time is 1 second.

To configure reporting of a MAC address move if the MAC address moves for a specified period of time, include the **notification-time** statement at the **[edit protocols l2-learning global-mac-move]** hierarchy level. The default notification timer is 1 second.

To configure reporting of a MAC address move if the MAC address moves a specified number of times, include the **threshold-count** statement at the **[edit protocols l2-learning global-mac-move]** hierarchy level. The default threshold count is 50 moves.

Use the **show l2-learning mac-move-buffer** command to view the actions as a result of MAC address move feature.

Use the **show l2-learning mac-move-buffer active** command to view the set of IFLs blocked as a result of MAC move action.

Use the **exclusive-mac** command exclude a MAC address from the MAC move limit algorithm, preventing a MAC address from being tracked.

Use the **clear l2-learning mac-move-buffer active** command to unblock the IFBDs that were blocked by MAC move action feature. This allows the user to keep the **reopen-time** configured to a large value, but when the looping error is fixed, user can manually release the blocking.

The following example sets the notification time for MAC moves to 1 second, the threshold time to 1 second, reopen-time to 180 seconds and the threshold count to 50 moves.

```
[edit protocols l2-learning]
global-mac-move {
   notification-time 1;
   reopen-time 180;
   threshold-count 50;
   threshold-time 1;
}
```

Related Documentation

- *Understanding Layer 2 Learning and Forwarding*
- *Configuring the MAC Table Timeout Interval*
- *Enabling MAC Accounting*
- *Limiting the Number of MAC Addresses Learned from Each Logical Interface*
- Disabling Layer 2 Learning and Forwarding on page 200

Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches and 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 "Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support (CLI Procedure)" on page 56. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

The Ethernet switching table, also known as the forwarding table, specifies the known locations of VLAN nodes. There are two ways to populate the Ethernet switching table on a switch. The easiest method is to let the switch update the table with MAC addresses.

The second way to populate the Ethernet switching table is to manually insert a VLAN node location into the table. You can do this to reduce flooding and speed up the switch's automatic learning process. To further optimize the switching process, indicate the next hop (next interface) packets will use after leaving the node.

Before configuring a static MAC address, be sure that you have:

- Set up the VLAN. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98 or"Configuring VLANs on Switches" on page 93.

To add a MAC address to the Ethernet switching table:

1. Specify the MAC address to add to the table:

```
[edit ethernet-switching-options]
set static vlan vlan-name mac mac-address
```

2. Indicate the next hop MAC address for packets sent to the indicated MAC address:

```
[edit ethernet-switching-options]
set static vlan vlan-name mac mac-address next-hop interface
```

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84

Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches and Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure)" on page 55. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

The Ethernet switching table, also known as the forwarding table, specifies the known locations of VLAN nodes and the addresses of devices within those nodes. There are two ways to populate the Ethernet switching table on a switch. The easiest method is to let the switch update the table with MAC addresses.

The second way to populate the Ethernet switching table is to manually insert addresses into the table. You can do this to reduce flooding and speed up the switch's automatic learning process.

Before configuring a static MAC address, be sure that you have:

- Set up the VLAN. See "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102.

To configure an interface to have a static MAC address:

```
[edit vlans vlan-name switch-options interface interface-name]
user@switch# set static-mac mac-address
```

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84

Example: Configuring the Default Learning for Unknown MAC Addresses

This example shows how to configure the device to use only ARP requests to learn the outgoing interfaces for unknown destination MAC addresses.

Requirements

Before you begin, determine the MAC addresses and associated interfaces of the forwarding table. See "Layer 2 Learning and Forwarding for VLANs Overview" on page 27.

Overview

In this example, you configure the device to use only ARP queries without traceroute requests.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security flow ethernet-switching no-packet-flooding no-trace-route
```

Step-by-Step Procedure

To configure the device to use only ARP requests to learn unknown destination MAC addresses:

1. Enable the device.

   ```
   [edit]
   user@host# set security flow ethernet-switching no-packet-flooding no-trace-route
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Verification

To verify the configuration is working properly, enter the **show security flow** command.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Understanding Integrated Routing and Bridging on page 445

- Example: Configuring an IRB Interface on a Security Device on page 452

Configuring MAC Limiting (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches and QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see *Configuring MAC Limiting (CLI Procedure)*. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

This topic describes various ways of configuring a limitation on MAC addresses in packets that are received and forwarded by the switch.

For information on configuring an interface to automatically recover from a shutdown caused by MAC limiting, see *Configuring Autorecovery from the Disabled State on Secure or Storm Control Interfaces (CLI Procedure)*. If you do not configure the switch for autorecovery from the disabled condition, you can bring up the disabled interfaces by running the **clear ethernet-switching recovery-timeout** command.

The different ways of setting a MAC limit are described in the following sections:

- Limiting the Number of MAC Addresses Learned by an Interface on page 58
- Limiting the Number of MAC Addresses Learned by a VLAN on page 58

Limiting the Number of MAC Addresses Learned by an Interface

To secure a port, you can set the maximum number of MAC addresses that can be learned by an interface:

- Set the MAC limit on an interface, and specify an action that the switch takes after the specified limit is exceeded:

  ```
  [edit switch-options]
  user@switch# set interface interface-name interface-mac-limit limit packet-action
  action
  ```

  After you set a new MAC limit for the interface, the system clears existing entries in the MAC address forwarding table associated with the interface.

Limiting the Number of MAC Addresses Learned by a VLAN

To limit the number of MAC addresses learned by a VLAN, perform both of the following steps:

1. Set the maximum number of MAC addresses that can be learned by a VLAN, and specify an action that the switch takes after the specified limit is exceeded:

   ```
   [edit vlans]
   user@switch# set vlan-name switch-options mac-table-size limit packet-action
   action
   ```

2. Set the maximum number of MAC addresses that can be learned by one *or* all interfaces in the VLAN, and specify an action that the switch takes after the specified limit is exceeded:

   NOTE: If you specify a MAC limit and packet action for all interfaces in the VLAN *and* a specific interface in the VLAN, the MAC limit and packet action specified at the specific interface level takes precedence. Also, at the VLAN interface level, only the **drop** and **drop-and-log** options are supported.

   ```
   [edit vlans]
   user@switch# set vlan-name switch-options interface interface-name
   interface-mac-limit limit packet-action action

   [edit vlans]
   user@switch# set vlan-name switch-options interface-mac-limit limit packet-action
   action
   ```

   After you set new MAC limits for a VLAN by using the **mac-table-size** statement or for interfaces associated with a VLAN by using the **interface-mac-limit** statement, the system clears the corresponding existing entries in the MAC address forwarding table.

   NOTE: On a QFX Series Virtual Chassis, if you include the **shutdown** option at the **[edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level and issue the **commit** operation, the system generates a commit error. The system does not generate an error if you include the **shutdown** option at the **[edit switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level.

Related Documentation

- *Configuring Autorecovery from the Disabled State on Secure or Storm Control Interfaces (CLI Procedure)*
- *Configuring Persistent MAC Learning (CLI Procedure)*

CHAPTER 4

Configuring MAC Learning

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.

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

Related Documentation

- Introduction to the Media Access Control (MAC) Layer 2 Sublayer on page 51
- Overview of Layer 2 Networking on page 21

Disabling MAC Learning on QFX Switches

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 on Devices with ELS Support" on page 62.

- 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 61
- Example: Disabling MAC Learning on a Switch on page 65
- no-mac-learning on page 1053

Disabling MAC Learning on Devices with ELS Support

By default, MAC learning is globally enabled on all node. This topic describes how to disable MAC learning, as well as how to reenable and verify that MAC learning has been enabled or disabled.

NOTE: This task supports the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3 If your switch runs software that does not support ELS, see "Disabling MAC Learning on QFX Switches" on page 61.

Disabling dynamic MAC learning prevents a node from learning source and destination MAC addresses.

- To disable MAC learning:

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

- To enable MAC learning:

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

- To verify the status of MAC learning, 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 61
- Example: Disabling MAC Learning on Devices with ELS Support on page 66
- no-mac-learning on page 1053

Disabling MAC Learning in a VLAN on a QFX Switch

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 61

- Example: Disabling MAC Learning in a VLAN on a QFX Series Switch on page 67
- no-mac-learning on page 1053

Disabling MAC Learning for a VLAN or Logical Interface

You can disable MAC learning for all logical interfaces in a specified VLAN, or for a specific logical interface in a VLAN. Disabling dynamic MAC learning prevents the specified interfaces from learning source MAC addresses.

To disable MAC learning for all logical interfaces in a VLAN in a virtual switch, include the **no-mac-learning** statement at the **[edit vlans *vlan-name* switch-options]** hierarchy level:

```
[edit]
vlans {
  vlan-name {
    domain-type bridge;
    interface interface-name;
    switch-options {
      no-mac-learning;
    }
  }
}
```

To disable MAC learning for a specific logical interface in a VLAN, include the **no-mac-learning** statement at the **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy level.

```
[edit]
vlans {
  vlan-name {
    domain-type bridge;
    interface interface-name;
    switch-options {
      interface interface-name {
        no-mac-learning;
      }
    }
  }
}
```

NOTE: When you disable MAC learning, source MAC addresses are not dynamically learned, and any packets sent to these source addresses are flooded into the VLAN.

NOTE: When you gather interfaces into a VLAN, the **no-mac-learn-enable** statement at the **[edit interfaces *interface-name* ether-options ethernet-switch-profile]** hierarchy level is not supported. You must use the **no-mac-learning** statement at the **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy level to disable MAC learning on an interface in a VLAN.

NOTE: When MAC learning is disabled for a VPLS routing instance, traffic is not load balanced and only one of the equal-cost next hops is used.

Disabling MAC Learning for a Set of VLANs

You can disable MAC learning for a set of VLANs. Disabling dynamic MAC learning prevents the Layer 2 trunk port associated with the set of VLANs from learning source and destination MAC addresses. When you disable MAC learning, source MAC addresses are not dynamically learned, and any packets sent to these source addresses are flooded into the switch.

To disable MAC learning for a set of VLANs, include the **no-mac-learning** statement at the **[edit switch-options]** hierarchy level:

```
[edit switch-options]
no-mac-learning;
```

Example: Disabling MAC Learning on a Switch

By default, MAC learning is enabled on the QFX Series and EX4600. 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 on Devices with ELS Support" on page 66.

- 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 61
- Disabling MAC Learning on QFX Switches on page 61
- no-mac-learning on page 1053

Example: Disabling MAC Learning on Devices with ELS Support

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, QFX3600, EX4600, QFX5100, and QFX10002 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Disabling MAC Learning on a Switch" on page 65. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

- To disable MAC learning in a VLAN:

```
[edit]
user@switch# set vlans vlan10 switch-options interface xe-0/0/0.0 no-mac-learning
```

- To reenable MAC learning:

```
[edit] vlans vlan10 switch-options interface 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 61
- Disabling MAC Learning on Devices with ELS Support on page 62

Example: Disabling MAC Learning in a VLAN on a QFX Series Switch

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 61
- Disabling MAC Learning in a VLAN on a QFX Switch on page 63
- no-mac-learning on page 1053
- show ethernet-switching table on page 1243

CHAPTER 5

Configuring MAC Accounting

Enabling MAC Accounting

By default, MAC accounting is disabled. You can enable packet accounting either for a router or switch as a whole or for a specific VLAN. After you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned.

To enable MAC accounting, include the **global-mac-statistics** statement at the **[edit protocols l2-learning]** hierarchy level:

```
[edit protocols l2-learning]
global-mac-statistics;
```

Enabling MAC Accounting for a VLAN

By default, MAC accounting is disabled. You can enable packet counting for a VLAN. When you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned on the interfaces in the VLAN.

To enable MAC accounting for a VLAN, include the **mac-statistics** statement at the **[edit vlans *vlan-name* switch-options]** hierarchy level:

```
[edit vlans vlan-name switch-options]
mac-statistics;
```

Enabling MAC Accounting for a Set of VLANs

By default, MAC accounting is disabled. You can enable packet counting for a set of VLANs. After you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned on the trunk port associated with the set of VLANs.

To enable MAC accounting for a set of VLANs, include the **mac-statistics** statement at the **[edit switch-options]** hierarchy level:

```
[edit switch-options]
```

```
mac-statistics;
```

Verifying That MAC Accounting Is Working

Purpose Verify that MAC accounting is enabled and the system is counting packets and collecting statistics.

Action

1. Verify that MAC accounting is enabled.

```
user@switch> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
        SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN101             88:e0:f3:bb:07:f0   D,SE           -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
       SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN102             88:e0:f3:bb:07:f0   D,SE           -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
       SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags               interface
    VLAN103             88:e0:f3:bb:07:f0   D,SE           -    ae20.0
[...output truncated...]
```

2. Display MAC accounting statistics for all VLANs associated with an interface.

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

 Local interface: ae20.0, Index: 1039
   Broadcast packets:                  115
   Broadcast bytes  :                 6900
   Multicast packets:               395113
   Multicast bytes  :             61622869
   Flooded packets  :                    0
   Flooded bytes    :                    0
   Unicast packets  :                 1419
   Unicast bytes    :               117924
   Current MAC count:                    4 (Limit 8192)
[...output truncated...]
```

3. Display MAC accounting statistics for each address in the MAC address table.

```
user@switch> show ethernet-switching table extensive
MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
```

```
VLAN ID: 101
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,acct,kernel,in_ifbd
   Epoch: 6                          Sequence number: 13
   Learning mask: 0x00000020
MAC address used as destination:
Packet count:                     0  Byte count:                     0
MAC address used as source:
Packet count:                     9  Byte count:                  1116

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 102
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,acct,kernel,in_ifbd
   Epoch: 6                          Sequence number: 13
   Learning mask: 0x00000020
MAC address used as destination:
Packet count:                     0  Byte count:                     0
MAC address used as source:
Packet count:                     9  Byte count:                  1116

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 103
   Learning interface: ae20/0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,acct,kernel,in_ifbd
   Epoch: 6                          Sequence number: 13
   Learning mask: 0x00000020
MAC address used as destination:
Packet count:                     0  Byte count:                     0
MAC address used as source:
Packet count:                     9  Byte count:                  1116
[...output truncated...]
```

Meaning In the output for **show ethernet-switching table**, the MAC flag **SE** indicates that MAC accounting is enabled for VLANs 101, 102, and 103, which are all associated with the **default-switch** routing instance.

The output for **show ethernet-switching statistics** displays packet statistics and the current number of MAC addresses learned by the VLANs associated with aggregated Ethernet interface **ae20.0**.

The output for **show ethernet-switching table extensive** shows information for each address in the MAC address table. In particular, it displays the number of packets sent to and received by an interface, which is identified by a MAC address.

The output from the three commands demonstrates that MAC accounting is working properly. That is, MAC accounting is enabled on VLANs 101, 102, and 103, and as a result, you can view statistics for each of these VLANs, aggregated Ethernet interface **ae20.0**, and each MAC address.

Related Documentation

- Enabling MAC Accounting on page 69
- Enabling MAC Accounting for a VLAN on page 69

- Enabling MAC Accounting for a Set of VLANs on page 69

CHAPTER 6

Configuring MAC Notification

Understanding MAC Notification on EX Series Switches

Juniper Networks EX Series Switches track clients on a network by storing Media Access Control (MAC) addresses in the Ethernet switching table on the switch. When switches learn or unlearn a MAC address, 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 MIB controls MAC notification for the network management system. For general information on the MAC Notification MIB, see the *Junos OS Network Management Configuration Guide*.

The MAC notification interval defines how often these SNMP notifications are sent to the network management system. The MAC notification interval works by tracking all of the MAC address additions or removals on the switch over a period of time and then sending all of the tracked MAC address additions or removals to the network management server at the end of the interval. 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.

Enabling MAC notification allows users to monitor the addition and removal of MAC addresses from the Ethernet switching table remotely using a network management system. The advantage of setting a high MAC notification interval is that the amount of network traffic is reduced because updates are sent less frequently. The advantage of setting a low MAC notification interval is that the network management system is better synchronized with the switch.

MAC notification is disabled by default. When MAC notification is enabled, the default MAC notification interval is 30 seconds.

Related Documentation

- Configuring Non-ELS MAC Notification on page 74

- *Configuring SNMP (J-Web Procedure)*

Configuring Non-ELS MAC Notification

NOTE: This task uses Junos OS for EX Series switches that do not support Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring MAC Notification on Switches with ELS Support (CLI Procedure)" on page 75. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

When a switch learns or unlearns a MAC address, 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 on page 74
- Disabling MAC Notification on page 74
- Setting the MAC Notification Interval on page 75

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, the MAC notification interval is set to 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 76

Configuring MAC Notification on Switches with ELS Support (CLI Procedure)

NOTE: This task uses the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Non-ELS MAC Notification" on page 74 or "Configuring Non-ELS MAC Notification" on page 74. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

When a switch learns or unlearns a MAC address, 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 on page 75
- Disabling MAC Notification on page 76
- Setting the MAC Notification Interval on page 76

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 switch-options]
user@switch# set mac-notification
```

To enable MAC notification on the switch with any other MAC notification interval (here, the MAC notification interval is set to 60 seconds):

```
[edit switch-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 switch-options]
user@switch# delete mac-notification
```

To disable MAC notification on a specific interface (here, the interface is ge-0/0/3):

```
[edit switch-options]
user@switch# set interface ge-0/0/3 no-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 switch-options]
user@switch# set mac-notification notification-interval 5
```

Related Documentation

- Verifying That MAC Notification Is Working Properly on page 76

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 on a QFX Series switch or an EX4600, and also to verify the MAC notification interval setting:

```
user@switch> show ethernet-switching mac-notification
Notification Status: Enabled
Notification Interval: 60
Notifications Sent          : 0
Notifications Table Maxsize : 256
```

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 60 seconds.

To verify that MAC notification is enabled on an EX Series switch while also verifying the MAC notification interval setting:

```
user@switch> show ethernet-switching mac-notification
Notification Status: Enabled
Notification Interval: 30
```

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 Non-ELS MAC Notification on page 74
- Configuring MAC Notification on Switches with ELS Support (CLI Procedure) on page 75

CHAPTER 7

Configuring MAC Table Aging

Understanding MAC Table Aging

Juniper Networks EX Series Ethernet Switches store MAC addresses in the Ethernet switching table, also called the *MAC table*. When the aging time for a MAC address in the table expires, the address is removed.

If your switch runs Juniper Networks Junos operating system (Junos OS) for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style, you can configure the MAC table aging time on all VLANs on the switch. If your switch runs Junos OS that does not support ELS, you can configure the MAC table aging time on all VLANs on the switch or on specified VLANs, as well as configure aging time to be unlimited, either on all VLANs or on specified VLANs, so that MAC addresses never age out of the table.

To learn MAC addresses, the switch reads all packets that it detects on the LAN or on the local VLAN, looking for MAC addresses of sending nodes. It places these addresses into its Ethernet switching table, along with two other pieces of information—the interface on which the traffic was received and the time when the address was learned.

When the switch receives traffic on an interface, it searches the Ethernet switching table for the MAC address of the destination. If the MAC address is not found, the traffic is flooded out all of the other interfaces associated with the VLAN. For example, if traffic is received on an interface that is associated with VLAN v-10 and there is no entry in the Ethernet switching table for VLAN v-10 (the Ethernet switching table is organized by VLAN), then the traffic is flooded to all access and trunk interfaces that are members of VLAN v-10.

Flooding allows the switch to learn about destinations that are not yet in its Ethernet switching table. If a particular destination MAC address is not in the Ethernet switching table, the switch floods the traffic to all interfaces except the interface on which it was received. When the destination node receives the flooded traffic, it sends an acknowledgment packet back to the switch, allowing the switch to learn the MAC address of the node and to add the address to its Ethernet switching table.

The switch uses a mechanism called aging to keep the Ethernet switching table current. For each MAC address in the Ethernet switching table, the switch records a timestamp of when the information about the network node was learned. Each time the switch detects traffic from a MAC address that is in its Ethernet switching table, it updates the timestamp of that MAC address. A timer on the switch periodically checks the timestamp, and if the MAC address of a node is older than the value set, the switch removes that MAC address from the Ethernet switching table. This aging process ensures that the switch tracks only active MAC addresses on the network and that it is able to flush out from the Ethernet switching table MAC addresses that are no longer available.

You configure how long MAC addresses remain in the Ethernet switching table by:

- (On switches that run Junos OS with support for the ELS configuration style) Using the **global-mac-table-aging-time** statement in the **[edit protocols l2-learning]** hierarchy.
- (On switches that run Junos OS that does not support ELS) Using the **mac-table-aging-time** statement in either the **[edit ethernet-switching-options]** or the **[edit vlans]** hierarchy, depending on whether you want to configure it for the entire switch or only for specific VLANs.

For example, in a topology with EX switches that run Junos OS that does not support ELS, if you have a printer VLAN, you might choose to configure the aging time for that VLAN to be considerably longer than for other VLANs so that MAC addresses of printers on this VLAN age out less frequently. Because the MAC addresses remain in the table, even if a printer has been idle for some time before traffic arrives for it, the switch still finds the MAC address and does not need to flood the traffic to all other interfaces.

Similarly, in a data center environment where the list of servers connected to the switch is fairly stable, you might choose to increase MAC address aging time, or even set it to unlimited, to increase the efficiency of the utilization of network bandwidth by reducing flooding.

Related Documentation

- Configuring MAC Table Aging on Switches on page 81
- *Controlling Authentication Session Timeouts (CLI Procedure)*

Configuring MAC Table Aging on Switches

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.

The following example uses Junos OS for Junos OS for QFX3500 and QFX3600 switches with no support for the Enhanced Layer 2 Software (ELS) configuration style. 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
```

NOTE: This command applies to all VLANs configured for the switch. You cannot configure separate MAC table aging times for specific VLANs.

The following example uses Junos OS for QFX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. Use the **global-mac-table-aging-time** command to configure how long entries remain in the Ethernet switching table before expiring, as follows:

```
[edit protocols l2-learning]
user@switch# set global-mac-table-aging-time 200
```

NOTE: This command applies to all VLANs configured for the switch. You cannot configure separate MAC table aging times for specific VLANs.

The following example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style.

The Ethernet switching table (or MAC table) aging process ensures that the EX Series switch tracks only active MAC addresses on the network and is able to flush out MAC addresses that are no longer used.

You can configure the MAC table aging time, the maximum time that an entry can remain in the Ethernet Switching table before it *ages out*, on all VLANs on the switch. This setting can influence efficiency of network resource use by affecting the amount of traffic that is flooded to all interfaces because when traffic is received for MAC addresses no longer in the Ethernet switching table, the switch floods the traffic to all interfaces.

```
[edit]
user@switch# set protocols l2-learning global-mac-table-aging-time seconds
```

The following example uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style.

The Ethernet switching table (or MAC table) aging process ensures that the EX Series switch tracks only active MAC addresses on the network and is able to flush out MAC addresses that are no longer used.

You can configure the MAC table aging time, the maximum time that an entry can remain in the Ethernet Switching table before it "ages out," either on all VLANs on the switch or on particular VLANs. This setting can influence efficiency of network resource use by affecting the amount of traffic that is flooded to all interfaces because when traffic is received for MAC addresses no longer in the Ethernet switching table, the switch floods the traffic to all interfaces.

To configure the MAC table aging time on all VLANs on the switch:

```
[edit]
user@switch# set ethernet-switching-options mac-table-aging-time seconds
```

To configure the MAC table aging time on a VLAN:

```
[edit]
user@switch# set vlans vlan-name mac-table-aging-time seconds
```

NOTE: You can set the MAC table aging time to unlimited. If you specify the value as unlimited, entries are never removed from the table. Generally, use this setting only if the switch or the VLAN has a fairly static number of end devices; otherwise the table will eventually fill up. You can use this setting to minimize traffic loss and flooding that might occur when traffic arrives for MAC addresses that have been removed from the table.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182
- *Controlling Authentication Session Timeouts (CLI Procedure)*
- Understanding Bridging and VLANs on Switches on page 84
- Understanding Bridging and VLANs on Switches on page 84
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support on page 131
- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support on page 170
- Example: Setting Up Bridging with Multiple VLANs on page 141
- Example: Connecting an Access Switch to a Distribution Switch on page 161
- Example: Setting Up Bridging with Multiple VLANs on Switches on page 147

CHAPTER 8

Configuring Bridging and VLANs

Understanding Bridging and VLANs on Switches

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 using VLAN IDs 1 through 4094, while VLAN IDs 0 and 4095 are reserved by Junos OS and cannot be assigned.
- On a switch running non-ELS software, you can configure 4091 VLANs using VLAN IDs 1-4094.

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.

Junos OS switches support the TPID value 0x9100 for Q-in-Q on switches. In addition to the TPID EtherType value of 0x8100, EX Series switches that do 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.

NOTE: **Q-in-Q tunnelling is not supported on NFX150 devices.**

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 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 87.

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.

NOTE: LACP is not supported on NFX150 devices.

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 87.

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 not using 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 most switches running 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).

On an EX Series switch that runs Junos OS that supports ELS, the maximum number of VLAN members allowed on the switch is as follows:

- EX4300—24 times the maximum number of VLANs that the switch supports (vmember limit = *vlan max* * 24)
- EX3400—16 times the maximum number of VLANs that the switch supports (vmember limit = *vlan max* * 16)
- EX2300—8 times the maximum number of VLANs that the switch supports (vmember limit = *vlan max* * 8)

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.

NOTE: LAG is not supported on NFX150 devices.

A Virtual Chassis Fabric supports up to 512,000 vmembers. The number of vmembers is based on the number of VLANs, and the number of interfaces configured in each VLAN.

A Default VLAN Is Configured on Most Switches

Some switches running 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.

EX Series switches that run Junos OS with the ELS configuration style do not support a default VLAN. The following EX Series switches running Junos OS not supporting the ELS configuration style are not preconfigured to belong to **default** or any other VLAN:

- Modular switches, such as the EX8200 switches and EX6200 switches
- Switches that are part of a Virtual Chassis

The reason that these switches are not preconfigured is that the physical configuration in both situations is flexible. There is no way of knowing which line cards have been inserted in either the EX8200 switch or EX6200 switch. There is also no way of knowing which switches are included in the Virtual Chassis. Switch interfaces in these two cases must first be defined as Ethernet switching interfaces. After an interface is defined as an Ethernet switching interface, the default VLAN appears in the output from the ? help and other commands.

NOTE: When a Juniper Networks EX4500 Ethernet Switch, EX4200 Ethernet Switch, EX3300 Ethernet Switch, 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.

NOTE: You cannot configure a default VLAN on NFX150 devices.

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.

NOTE: Two logical interfaces that are configured on the same physical interface cannot be mapped to the same VLAN.

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)" on page 55.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)" on page 55.

For information about using 802.1X authentication to authenticate end devices and allow access to dynamic VLANs configured on a RADIUS server, see *Understanding Dynamic*

VLAN Assignment Using RADIUS Attributes. You can optionally implement this feature to offload the manual assignment of VLAN traffic to automated RADIUS server databases.

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 87.

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.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- *Understanding FCoE*
- *Interfaces Overview for Switches*
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496
- Understanding Integrated Routing and Bridging on page 445

Configuring VLANs on Switches

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 on Switches with Enhanced Layer 2 Support" on page 97.

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 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 Switches on page 104
- Configuring IRB Interfaces on Switches on page 454
- Creating a Series of Tagged VLANs on page 214
- Understanding Bridging and VLANs on Switches on page 84

Configuring Integrated Routing and Bridging for VLANs

Integrated routing and bridging (IRB) provides simultaneous support for Layer 2 bridging and Layer 3 routing on the same interface. IRB enables you to route packets to another routed interface or to another VLAN that has an IRB interface configured. You configure a logical routing interface by specifying **irb** as an interface name at the **[edit interfaces]** hierarchy level and including that interface in the VLAN.

NOTE: You can include only one Layer 3 interface in a VLAN.

To configure a VLAN with IRB support, include the following statements:

```
[edit]
vlans {
  vlan-name {
    domain-type bridge;
    interface interface-name;
    l3-interface (VLAN) interface-name;
    vlan-id (none | number);
    vlan-tags outer number inner number;
  }
}
```

For each VLAN that you configure, specify a ***vlan-name***. You must also specify the value **bridge** for the **domain-type** statement.

For the **vlan-id** statement, you can specify either a valid VLAN identifier or the **none** option.

NOTE: If you configure a Layer 3 interface to support IRB in a VLAN, you cannot use the **all** option for the **vlan-id** statement.

The **vlan-tags** statement enables you to specify a pair of VLAN identifiers; an **outer** tag and an **inner** tag.

NOTE: For a single VLAN, you can include either the **vlan-id** statement or the **vlan-tags** statement, but not both.

To include one or more logical interfaces in the VLAN, specify the ***interface-name*** for each Ethernet interface to include that you configured at the **[edit interfaces]** hierarchy level.

NOTE: A maximum of 4096 active logical interfaces are supported for a VLAN or on each mesh group in a VPLS routing instance configured for Layer 2 bridging.

To associate a Layer 3 interface with a VLAN, include the **l3-interface** ***interface-name*** statement and specify an ***interface-name*** you configured at the **[edit interfaces irb]** hierarchy level. You can configure only one Layer 3 interface for each VLAN.

IRB interfaces are supported for multicast snooping.

In multihomed VPLS configurations, you can configure VPLS to keep a VPLS connection up if only an IRB interface is available by configuring the **irb** option for the **connectivity-type** statement at the **[edit routing-instances** ***routing-instance-name*** **protocols vpls]** hierarchy level. The **connectivity-type** statement has the **ce** and **irb** options. The **ce** option is the default and specifies that a CE interface is required to maintain the VPLS connection. By default, if only an IRB interface is available, the VPLS connection is brought down.

NOTE: When you configure IRB interfaces in more than one logical system on a device, all of the IRB logical interfaces share the same MAC address.

Configuring a Layer 2 Virtual Switch on an EX Series Switch

A Layer 2 virtual switch, which isolates a LAN segment with its spanning-tree protocol instance and separates its VLAN ID space, filters and forwards traffic only at the data link layer. Each VLAN consists of a set of logical ports that participate in Layer 2 learning and forwarding. A virtual switch represents a Layer 2 network.

Two main types of interfaces are used in virtual switch hierarchies:

- Layer 2 logical interface—This type of interface uses the VLAN-ID as a virtual circuit identifier and the scope of the VLAN-ID is local to the interface port. This type of interface is often used in service-provider-centric applications.
- Access or trunk interface—This type of interface uses a VLAN-ID with global significance. The access or trunk interface is implicitly associated with VLANs based on VLAN membership. Access or trunk interfaces are typically used in enterprise-centric applications.

NOTE: The difference between access interfaces and trunk interfaces is that access interfaces can be part of one VLAN only and the interface is normally attached to an end-user device (packets are implicitly associated with the configured VLAN). In contrast, trunk interfaces multiplex traffic from multiple VLANs and usually interconnect switches.

To configure a Layer 2 virtual switch, include the following statements:

```
[edit]
routing-instances {
  routing-instance-name (
    instance-type virtual-switch;
    vlans vlan-name{
      vlan-id (all | none | number);
        [...configure optional VLAN parameters]
    }
  }
}
```

To enable a virtual switch, you must specify **virtual-switch** as the **instance-type**.

The VLANs that are specified with the **vlan-id** statement are included in the virtual switch.

You can configure other optional VLAN parameters in the virtual switch.

Related Documentation

- Configuring a Layer 2 Virtual Switch with a Layer 2 Trunk Port on page 96
- *Configuring a Layer 2 Virtual Switch*

Configuring a Layer 2 Virtual Switch with a Layer 2 Trunk Port

You can associate one or more Layer 2 trunk interfaces with a virtual switch.

A virtual switch configured with a Layer 2 trunk port also supports IRB within a VLAN. IRB provides simultaneous support for Layer 2 bridging and Layer 3 IP routing on the same interface. Only an interface configured with the **interface-mode (access | trunk)** statement can be associated with a virtual switch. An access interface enables you to accept packets with no VLAN identifier.

In addition, you can configure Layer 2 learning and forwarding properties for the virtual switch.

To configure a virtual switch with a Layer 2 trunk interface, include the following statements:

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type virtual-switch;
    interface interface-name;
    vlans name{
      vlan-id (all | none | number);
        [...configure optional VLAN parameters]
    }
  }
}
```

Related Documentation

- Configuring a Layer 2 Virtual Switch on an EX Series Switch on page 95

Configuring VLANs on Switches with Enhanced Layer 2 Support

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 supports the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3. If your switch runs software that does not support ELS, see "Configuring VLANs on Switches" on page 93.

NOTE: Starting with Junos OS Release 17.1R3, on QFX10000 switches, you cannot configure an interface with both **family ethernet-switching** and **flexible-vlan-tagging**. This configuration is not supported, and a warning will be issued if you try to commit this configuration.

NOTE: Two logical interfaces that are configured on the same physical interface cannot be mapped to the same VLAN.

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: Switches that run Junos OS with the ELS configuration style do not support a default VLAN. 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.

NOTE: On QFX5100 switches running Junos OS Release 14.1X53-D46 or earlier, when you configure an interface under a VLAN but do not specify the name of the VLAN, the system will not issue a commit error.

```
[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
```

NOTE: The **family inet** option is not supported on NFX150 devices.

4. Configure the VLAN tag ID or VLAN ID list for the VLAN:

```
[edit vlans]
user@switch# set vlan-name vlan-id vlan-id-number
```

or

```
[edit vlans]
user@switch# set vlan-name vlan-id-list [vlan-ids | vlan-id--vlan-id-]
```

5. 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
```

Release History Table

Release	Description
17.1R3	Starting with Junos OS Release 17.1R3, on QFX10000 switches, you cannot configure an interface with both **family ethernet-switching** and **flexible-vlan-tagging**.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Configuring IRB Interfaces on Switches on page 454
- Understanding Bridging and VLANs on Switches on page 84

Configuring VLANs for EX Series Switches (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

EX Series switches use VLANs to make logical groupings of network nodes with their own broadcast domains. VLANs limit the traffic flowing across the entire LAN and reduce collisions and packet retransmissions.

Why Create a VLAN?

Some reasons to create VLANs are:

- A LAN has more than 200 devices.
- A LAN has a large amount of broadcast traffic.
- A group of clients requires that a higher-than-average level of security be applied to traffic entering or exiting the group's devices.
- A group of clients requires that the group's devices receive less broadcast traffic than they are currently receiving, so that data speed across the group is increased.

Create a VLAN Using the Minimum Procedure

Two steps are required to create a VLAN:

- Uniquely identify the VLAN. You do this by assigning either a name or an ID (or both) to the VLAN. When you assign just a VLAN name, an ID is generated by Junos OS.
- Assign at least one switch port interface to the VLAN for communication. All interfaces in a single VLAN are in a single broadcast domain, even if the interfaces are on different switches. You can assign traffic on any switch to a particular VLAN by referencing either the interface sending traffic or the MAC addresses of devices sending traffic.

The following example creates a VLAN using only the two required steps. The VLAN is created with the name employee-vlan. Then, three interfaces are assigned to that VLAN so that the traffic is transmitted among these interfaces.

NOTE: In this example, you could alternatively assign an ID number to the VLAN. The requirement is that the VLAN have a unique ID.

```
[edit] set vlans employee-vlan
[edit] set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
[edit] set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members
  employee-vlan
[edit] set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members
  employee-vlan
```

In the example, all users connected to the interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 can communicate with each other, but not with users on other interfaces in this network.

To configure communication between VLANs, you must configure a routed VLAN interface (RVI). See "Configuring Routed VLAN Interfaces on Switches (CLI Procedure)" on page 365.

Create a VLAN Using All of the Options

To configure a VLAN, follow these steps:

1. In configuration mode, create the VLAN by setting the unique VLAN name:

   ```
   [edit]user@switch# set vlans vlan-name
   ```

2. Configure the VLAN tag ID or VLAN ID range for the VLAN. (If you assigned a VLAN name, you do not have to do this, because a VLAN ID is assigned automatically, thereby associating the name of the VLAN to an ID number. However, if you want to control the ID numbers, you can assign both a name and an ID.)

   ```
   [edit]user@switch# set vlans vlan-name vlan-id vlan-id-number
   ```

   or

   ```
   [edit]user@switch# set vlans vlan-name vlan-range (vlan-id-low) - (vlan-id-high)
   ```

3. Assign at least one interface to the VLAN:

   ```
   [edit]user@switch# set vlans vlan-name interface interface-name
   ```

   NOTE: You can also specify that a trunk interface is 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.

4. (Optional) Create a subnet for the VLAN because all computers that belong to a subnet are addressed with a common, identical, most-significant-bit group in their IP address. This makes it easy to identify VLAN members by their IP addresses. To create the subnet for the VLAN:

   ```
   [edit interfaces]user@switch# set vlan unit logical-unit-number family inet address ip-address
   ```

5. (Optional) Specify the description of the VLAN:

   ```
   [edit]user@switch# set vlans vlan-name description text-description
   ```

6. (Optional) To avoid exceeding the maximum number of members allowed in a VLAN, specify the maximum time that an entry can remain in the forwarding table before it ages out:

   ```
   [edit]user@switch# set vlans vlan-name mac-table-aging-time time
   ```

7. (Optional) For security purposes, specify a VLAN firewall filter to be applied to incoming or outgoing packets:

   ```
   [edit]user@switch# set vlans vlan-name filter input-or-output filter-name
   ```

8. (Optional) For accounting purposes, enable a counter to track the number of times this VLAN is accessed:

   ```
   [edit]user@switch# set vlans vlan-name l3-interface ingress-counting l3-interface-name
   ```

9. (Optional) For Virtual Chassis bandwidth management purposes, enable VLAN Pruning to ensure all broadcast, multicast, and unknown unicast traffic entering the Virtual Chassis on the VLAN uses the shortest possible path through the Virtual Chassis:

   ```
   [edit]
   user@switch# set vlans vlan-name vlan-prune
   ```

Configuration Guidelines for VLANs

Two steps are required to create a VLAN. You must uniquely identify the VLAN and you must assign at least one switch port interface to the VLAN for communication.

After creating a VLAN, all users all users connected to the interfaces assigned to the VLAN can communicate with each other but not with users on other interfaces in the network. To configure communication between VLANs, you must configure a routed VLAN interface (RVI). See "Configuring Routed VLAN Interfaces on Switches (CLI Procedure)" on page 365 to create an RVI.

The number of VLANs supported per switch varies for each switch type. Use the command **set vlans id vlan-id ?** to discover the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because each VLAN is assigned an ID number when it is created. You can, however, exceed the recommended VLAN member maximum . To determine the maximum number of VLAN members allowed on a switch, multiply the VLAN maximum obtained using **set vlans id vlan-id ?** times 8.

If a switch configuration exceeds the recommended VLAN member maximum, you see a warning message when you commit the configuration. If you ignore the warning and commit such a configuration, the configuration succeeds but you run the risk of crashing the Ethernet switching process (eswd) due to memory allocation failure.

NOTE: When EX2300 and EX3400 ERPS switches have a VLAN-ID configured with a name under an interface hierarchy, a commit error occurs. Avoid this by configuring VLAN-IDs using numbers when they are under an interface hierarchy with ERPS configured in the switch.

Related Documentation

- *Configuring VLANs for EX Series Switches (J-Web Procedure)*
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Creating a Series of Tagged VLANs on EX Series Switches (CLI Procedure) on page 218

- Understanding Bridging and VLANs on Switches on page 84
- Understanding Integrated Routing and Bridging on page 445

Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

EX Series switches use VLANs to make logical groupings of network nodes with their own broadcast domains. VLANs limit the traffic flowing across the entire LAN and reduce collisions and packet retransmissions.

- Why Create a VLAN? on page 102
- Creating a VLAN Using the Minimum Procedure on page 102
- Creating a VLAN Using All of the Options on page 103
- Configuration Guidelines for VLANs on page 104

Why Create a VLAN?

For switching to begin, you must explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist.

Some reasons to create more than one VLAN are:

- A LAN has more than 200 devices.
- A LAN has a large amount of broadcast traffic.
- A group of clients requires that a higher-than-average level of security be applied to traffic entering or exiting the group's devices.
- A group of clients requires that the group's devices receive less broadcast traffic than they are currently receiving, so that data speed across the group is increased.

Creating a VLAN Using the Minimum Procedure

These steps are required to create a VLAN:

- Uniquely identify the VLAN. You do this by assigning a name and an ID to the VLAN.
- Assign at least one switch port interface to the VLAN for communication. After assigning one or more interfaces to the VLAN, the interfaces function in access mode. All interfaces in a single VLAN are in a single broadcast domain, even if the interfaces are on different switches. You can assign traffic on any switch to a particular VLAN by referencing either the interface sending traffic or the MAC addresses of devices sending traffic.

The following example creates a VLAN using only a few required steps. The VLAN is created with the name **employee-vlan** and the VLAN ID of **100**. Then, three interfaces are assigned to that VLAN, and these interfaces, which function in access mode, transmit traffic among themselves.

```
[edit] set vlans employee-vlan
[edit] set vlans employee-vlan vlan-id 100
[edit] set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
 employee-vlan
[edit] set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members
 employee-vlan
[edit] set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members
 employee-vlan
```

In the example, all users connected to the interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 can communicate with each other, but not with users on other interfaces in this network. To configure communication between VLANs, you must configure an integrated routing and bridging (IRB) interface. See "Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure)" on page 456.

Creating a VLAN Using All of the Options

To configure a VLAN, follow these steps:

1. Create the VLAN by setting the unique VLAN name:

   ```
   [edit vlans]
   user@switch# set vlan-name
   ```

2. Configure the VLAN ID or a VLAN ID list for the VLAN. Using the VLAN ID list option, you can optionally specify a range of VLAN IDs.

   ```
   [edit vlans]
   user@switch# set vlan-name vlan-id vlan-id-number
   ```

   or

   ```
   [edit vlans]
   user@switch# set vlan-name vlan-id-list [vlan-ids | vlan-id--vlan-id-]
   ```

3. Assign at least one interface to the VLAN:

   ```
   [edit interfaces]
   user@switch# set interface-name unit logical-unit-number family ethernet-switching vlan
   members [all | vlan-names | vlan-ids]
   ```

   NOTE: You can also specify that a trunk interface is a member of all 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.

4. (Optional) Create a subnet for the VLAN because all computers that belong to a subnet are addressed with a common, identical, most-significant-bit group in their IP

address. This makes it easy to identify VLAN members by their IP addresses. To create the subnet for the VLAN:

```
[edit interfaces]
user@switch# set vlan unit logical-unit-number family inet address
ip-address/destination-prefix
```

5. (Optional) Specify the description of the VLAN:

```
[edit vlans]
user@switch# set vlan-name description text-description
```

6. (Optional) For security purposes, 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
```

Configuration Guidelines for VLANs

To create a VLAN, you must uniquely identify the VLAN and assign at least one switch port interface to the VLAN for communication. After you assign one or more interfaces to the VLAN, the interfaces function in access mode.

After creating a VLAN, all users connected to interfaces that are assigned to the VLAN can communicate with each other but not with users on other interfaces in the network. To configure communication between VLANs, you must configure an IRB interface. For information about creating an IRB interface, see "Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure)" on page 456..

The number of VLANs supported per switch varies. Use the 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 each VLAN is assigned an ID number when it is created.

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84
- Understanding Integrated Routing and Bridging on page 445
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support on page 131

Example: Setting Up Basic Bridging and a VLAN on Switches

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.

NOTE: You cannot configure more than one logical interface that belongs to the same physical interface in the same bridge domain.

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 35: 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**

Table 35: Components of the Basic Bridging Configuration Topology (continued)

Property	Settings
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 115
- Verifying That Interfaces Are Associated with the Proper VLANs on page 116

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 141
- Understanding Bridging and VLANs on Switches on page 84

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch

NOTE: This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support" on page 131. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3

EX Series switches use bridging and virtual LANs (VLANs) to connect network devices in a LAN—desktop computers, IP telephones, printers, file servers, wireless access points, and others—and to segment the LAN into smaller bridging domains. The switch's default configuration provides a quick setup of bridging and a single VLAN.

This example describes how to configure basic bridging and VLANs for an EX Series switch:

Requirements

This example uses the following software and hardware components:

- Junos OS Release 9.0 or later for EX Series switches
- One EX4200 Virtual Chassis switch

Before you set up bridging and a VLAN, be sure you have:

- Installed your EX Series switch. See *Installing and Connecting an EX3200 Switch*.
- Performed the initial switch configuration. See *Connecting and Configuring an EX Series Switch (J-Web Procedure)*.

Overview and Topology

EX Series switches connect network devices in an office LAN or a data center LAN to provide sharing of common resources such as printers and file servers and to enable wireless devices to connect to the LAN through wireless access points. 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. 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.

To use an EX Series switch to connect network devices on a LAN, you must, at a minimum, configure bridging and VLANs. If you simply power on the switch and perform the initial switch configuration using the factory-default settings, bridging is enabled on all the switch's interfaces, all interfaces are in access mode, and all interfaces belong to a VLAN called **default**, which is automatically configured. When you plug access devices—such as desktop computers, Avaya IP telephones, file servers, printers, and wireless access points—into the switch, they are joined immediately into the **default** VLAN and the LAN is up and running.

The topology used in this example consists of one EX4200-24T switch, which has a total of 24 ports. Eight of the ports support Power over Ethernet (PoE), which means they provide both network connectivity and electric power for the device connecting to the

port. To these ports, you can plug in devices requiring PoE, such as Avaya VoIP telephones, wireless access points, and some IP cameras. (Avaya phones have a built-in hub that allows you to connect a desktop PC to the phone, so the desktop and phone in a single office require only one port on the switch.) The remaining 16 ports provide only network connectivity. You use them to connect devices that have their own power sources, such as desktop and laptop computers, printers, and servers. Table 36 on page 124 details the topology used in this configuration example.

Table 36: Components of the Basic Bridging Configuration Topology

Property	Settings
Switch hardware	EX4200-24T switch, with 24 Gigabit Ethernet ports: 8 PoE ports (**ge-0/0/0** through **ge-0/0/7**) and 16 non-PoE ports (**ge-0/0/8** through **ge-0/0/23**)
VLAN name	**default**
Connection to wireless access point (requires PoE)	**ge-0/0/0**
Connections to Avaya IP telephone—with integrated hub, to connect phone and desktop PC to a single port (requires PoE)	**ge-0/0/1** through **ge-0/0/7**
Direct connections to desktop PCs (no PoE required)	**ge-0/0/8** through **ge-0/0/12**
Connections to file servers (no PoE required)	**ge-0/0/17** and **ge-0/0/18**
Connections to integrated printer/fax/copier machines (no PoE required)	**ge-0/0/19** through **ge-0/0/20**
Unused ports (for future expansion)	**ge-0/0/13** through **ge-0/0/16**, and **ge-0/0/21** through **ge-0/0/23**

Configuration

CLI Quick Configuration

By default, after you perform the initial configuration on the EX4200 switch, switching is enabled on all interfaces, a VLAN named **default** is created, and all interfaces are placed into this VLAN. You do not need to perform any other configuration on the switch to set up bridging and VLANs. To use the switch, simply plug the Avaya IP phones into the PoE-enabled ports **ge-0/0/1** through **ge-0/0/7**, and plug in the PCs, file servers, and printers to the non-PoE ports, **ge-0/0/8** through **ge-0/0/12** and **ge-0/0/17** through **ge-0/0/20**.

Step-by-Step Procedure

To configure bridging and VLANs:

1. Make sure the switch is powered on.
2. Connect the wireless access point to switch port **ge-0/0/0**.
3. Connect the seven Avaya phones to switch ports **ge-0/0/1** through **ge-0/0/7**.

4. Connect the five PCs to ports **ge-0/0/8** through **ge-0/0/12**.

5. Connect the two file servers to ports **ge-0/0/17** and **ge-0/0/18**.

6. Connect the two printers to ports **ge-0/0/19** and **ge-0/0/20**.

Results Check the results of the configuration:

```
user@switch> show configuration
## Last commit: 2008-03-06 00:11:22 UTC by triumph
version 9.0;
system {
  root-authentication {
    encrypted-password "$1$urmA7AFM$x5SaGEUOdSI3u1K/iITGh1"; ## SECRET-DATA
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any notice;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
  commit {
    factory-settings {
      reset-chassis-lcd-menu;
      reset-virtual-chassis-configuration;
    }
  }
}
interfaces {
  ge-0/0/0 {
    unit 0 {
      family ethernet-switching;
    }
  }
  ge-0/0/1 {
    unit 0 {
      family ethernet-switching;
    }
  }
  ge-0/0/2 {
    unit 0 {
      family ethernet-switching;
    }
  }
  ge-0/0/3 {
    unit 0 {
      family ethernet-switching;
```

```
    }
}
ge-0/0/4 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/5 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/6 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/7 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/8 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/9 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/10 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/11 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/12 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/13 {
    unit 0 {
        family ethernet-switching;
    }
}
ge-0/0/14 {
    unit 0 {
        family ethernet-switching;
    }
```

```
}
ge-0/0/15 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/16 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/17 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/18 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/19 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/20 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/21 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/22 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/0/23 {
  unit 0 {
    family ethernet-switching;
  }
}
ge-0/1/0 {
  unit 0 {
    family ethernet-switching;
  }
}
xe-0/1/0 {
  unit 0 {
    family ethernet-switching;
  }
}
```

```
    ge-0/1/1 {
      unit 0 {
        family ethernet-switching;
      }
    }
    xe-0/1/1 {
      unit 0 {
        family ethernet-switching;
      }
    }
    ge-0/1/2 {
      unit 0 {
        family ethernet-switching;
      }
    }
    ge-0/1/3 {
      unit 0 {
        family ethernet-switching;
      }
    }
  }
  protocols {
    lldp {
      interface all;
    }
    rstp;
  }
    poe {
      interface all;
    }
```

Verification

To verify that switching is operational and that a VLAN has been created, perform these tasks:

- Verifying That the VLAN Has Been Created on page 128
- Verifying That Interfaces Are Associated with the Proper VLANs on page 129

Verifying That the VLAN Has Been Created

Purpose Verify that the VLAN named **default** has been created on the switch.

Action List all VLANs configured on the switch:

```
user@switch> show vlans

Name            Tag     Interfaces
default
                        ge-0/0/0.0*, 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-0/1/0.0*, ge-0/1/1.0*, ge-0/1/2.0*, ge-0/1/3.0*
mgmt
                        me0.0*
```

Meaning The **show vlans** command lists the VLANs configured on the switch. This output shows that the VLAN **default** 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

Interface   State    VLAN members           Blocking
ge-0/0/0.0  up       default                unblocked
ge-0/0/1.0  down     default                blocked - blocked by STP/RTG
ge-0/0/2.0  down     default                blocked - blocked by STP/RTG
ge-0/0/3.0  down     default                blocked - blocked by STP/RTG
ge-0/0/4.0  down     default                blocked - blocked by STP/RTG
ge-0/0/5.0  down     default                blocked - blocked by STP/RTG
ge-0/0/6.0  down     default                blocked - blocked by STP/RTG
ge-0/0/7.0  down     default                blocked - blocked by STP/RTG
ge-0/0/8.0  up       default                unblocked
ge-0/0/9.0  down     default                blocked - blocked by STP/RTG
ge-0/0/10.0 down     default                blocked - blocked by STP/RTG
ge-0/0/11.0 up       default                unblocked
ge-0/0/12.0 down     default                blocked - blocked by STP/RTG
ge-0/0/13.0 down     default                blocked - blocked by STP/RTG
ge-0/0/14.0 down     default                blocked - blocked by STP/RTG
ge-0/0/15.0 down     default                blocked - blocked by STP/RTG
ge-0/0/16.0 down     default                blocked - blocked by STP/RTG
ge-0/0/17.0 down     default                blocked - blocked by STP/RTG
ge-0/0/18.0 down     default                blocked - blocked by STP/RTG
ge-0/0/19.0 up       default                unblocked
ge-0/0/20.0 down     default                blocked - blocked by STP/RTG
ge-0/0/21.0 down     default                blocked - blocked by STP/RTG
ge-0/0/22.0 down     default                blocked - blocked by STP/RTG
ge-0/0/23.0 down     default                blocked - blocked by STP/RTG
ge-0/1/0.0  up       default                unblocked
ge-0/1/1.0  up       default                unblocked
ge-0/1/2.0  up       default                unblocked
ge-0/1/3.0  up       default                unblocked
me0.0       up       mgmt                   unblocked
```

Meaning The **show ethernet-switching interfaces** command lists all interfaces on which switching is enabled (in the **Interfaces** 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, **ge-0/0/0** through **ge-0/0/12** and **ge-0/0/17** through **ge-0/0/20** and that they are all part of VLAN **default**. Notice that the interfaces listed are the logical interfaces, not the physical interfaces. For example, the output shows **ge-0/0/0.0** instead of **ge-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 for EX Series Switches on page 153
- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182
- Understanding Bridging and VLANs on Switches on page 84

Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support

NOTE: This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs Junos OS that does not support ELS, see "Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch" on page 122. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

EX Series switches use bridging and virtual LANs (VLANs) to connect network devices in a LAN—desktop computers or laptops, IP telephones, printers, file servers, wireless access points, and others—and to segment the LAN into smaller broadcast domains.

This example describes how to configure basic bridging and a VLAN on an EX Series switch:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 13.2X50-D10 or later for EX Series switches

Before you set up bridging and a VLAN, be sure you have:

- Installed your EX Series switch. See the installation instructions for your switch.
- Performed the initial switch configuration. See *Connecting and Configuring an EX Series Switch (CLI Procedure)*.

Overview and Topology

EX Series switches connect network devices in an office LAN or a data center LAN to provide sharing of common resources such as printers and file servers and to enable wireless devices to connect to the LAN through wireless access points. 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. 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.

To use an EX Series switch to connect network devices on a LAN, you must, at a minimum, explicitly configure at least one VLAN, even if your network is simple and you want only one broadcast domain to exist, as is the case with this example. You must also assign all needed interfaces to the VLAN, after which the interfaces function in access mode. After the VLAN is configured, you can plug access devices—such as desktop or laptop computers, IP telephones, file servers, printers, and wireless access points—into the switch, and they are joined immediately into the VLAN, and the LAN is up and running.

The topology used in this example consists of one EX4300-24P switch, which has a total of 24 ports. All ports support Power over Ethernet (PoE), which means they provide both network connectivity and electric power for the device connecting to the port. To these ports, you can plug in devices requiring PoE, such as Avaya VoIP telephones, wireless access points, and some IP cameras. (Avaya phones have a built-in hub that allows you to connect a desktop PC to the phone, so the desktop and phone in a single office require only one port on the switch.) Table 36 on page 124 details the topology used in this configuration example.

Table 37: Components of the Basic Bridging Configuration Topology

Property	Settings
Switch hardware	EX4300-24P switch, with 24 Gigabit Ethernet ports: in this example, 8 ports are used as PoE ports (ge-0/0/0 through ge-0/0/7) and 16 ports used as non-PoE ports (ge-0/0/8 through ge-0/0/23)
VLAN name	employee-vlan
VLAN ID	10
Connection to wireless access point (requires PoE)	ge-0/0/0
Connections to Avaya IP telephone—with integrated hub, to connect phone and desktop PC to a single port (requires PoE)	ge-0/0/1 through ge-0/0/7
Direct connections to desktop PCs and laptops (no PoE required)	ge-0/0/8 through ge-0/0/12
Connections to file servers (no PoE required)	ge-0/0/17 and ge-0/0/18
Connections to integrated printer/fax/copier machines (no PoE required)	ge-0/0/19 through ge-0/0/20
Unused ports (for future expansion)	ge-0/0/13 through ge-0/0/16, and ge-0/0/21 through ge-0/0/23

Configuration

To set up basic bridging and a VLAN:

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 ge-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
set interfaces ge-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
```

You must then plug the wireless access point into PoE-enabled port **ge-0/0/0** and the Avaya IP phones into the PoE-enabled ports **ge-0/0/1** through **ge-0/0/7**. Also, plug the PCs, file servers, and printers into ports **ge-0/0/8** through **ge-0/0/12** and **ge-0/0/17** through **ge-0/0/20**.

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 ge-0/0/0 through ge-0/0/12, and ge-0/0/17 through ge-0/0/20 to the employee-vlan VLAN:

   ```
   [edit interface]
   user@switch# set ge-0/0/0 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/1 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/2 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/3 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/4 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/5 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/6 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/7 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/8 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/9 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/10 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/11 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/12 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/17 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/18 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/19 unit 0 family ethernet-switching vlan members employee-vlan
   user@switch# set ge-0/0/20 unit 0 family ethernet-switching vlan members employee-vlan
   ```

3. Connect the wireless access point to switch port ge-0/0/0.
4. Connect the seven Avaya phones to switch ports ge-0/0/1 through ge-0/0/7.
5. Connect the five PCs to ports ge-0/0/8 through ge-0/0/12.
6. Connect the two file servers to ports ge-0/0/17 and ge-0/0/18.
7. Connect the two printers to ports ge-0/0/19 and ge-0/0/20.

Results Check the results of the configuration:

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

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

```
ge-0/0/17 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/18 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/19 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members employee-vlan;
            }
        }
    }
}
ge-0/0/20 {
    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 137
- Verifying That Interfaces Are Associated with the Proper VLANs on page 138

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
                                                           ge-0/0/0.0
                                                           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-0/0/17.0
                                                           ge-0/0/18.0
                                                           ge-0/0/19.0
                                                           ge-0/0/20.0
...
```

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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-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
ge-0/0/20.0                   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, ge-0/0/0 through ge-0/0/12 and ge-0/0/17 through ge-0/0/20 and that they 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 ge-0/0/0.0 instead of ge-0/0/0. This is because Junos OS creates VLANs on logical interfaces, not directly on physical interfaces.

Related Documentation

- Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure) on page 102
- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support on page 170
- Understanding Bridging and VLANs on Switches on page 84

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 on Switches" on page 147.

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 38: 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 38: 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 146
- Verifying That Traffic Is Being Routed Between the Two VLANs on page 146
- Verifying That Traffic Is Being Switched Between the Two VLANs on page 147

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 Switches on page 104

Example: Setting Up Bridging with Multiple VLANs on Switches

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.

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

NOTE: This task supports the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3. If your switch runs software that does not supports ELS, see "Example: Setting Up Bridging with Multiple VLANs" on page 141.

Requirements

This example uses the following hardware and software components:

- A configured and provisioned QFX3500 switch
- Junos OS Release 13.2X50-D15 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 39: 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 39: 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 irb.0
set vlans sales vlan-id 100
set vlans support vlan-id 200
set vlans support l3-interface irb.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 irb.0
user@switch# set support l3-interface irb.1
```

Configuration Results

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 irb0;
      }
      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 irb1;
      }
    }
```

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 152
- Verifying That Traffic Is Being Routed Between the Two VLANs on page 152
- Verifying That Traffic Is Being Switched Between the Two VLANs on page 153

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 Switches on page 104
- Understanding Bridging and VLANs on Switches on page 84

Example: Setting Up Bridging with Multiple VLANs for EX Series Switches

To segment traffic on a LAN into separate broadcast domains, you create separate virtual LANs (VLANs) on an EX Series 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 bridging for an EX Series switch and how to create two VLANs to segment the LAN:

Requirements

This example uses the following hardware and software components:

- One EX4200-48P Virtual Chassis switch
- Junos OS Release 9.0 or later for EX Series switches

Before you set up bridging and VLANs, be sure you have:

- Installed the EX Series switch. See *Installing and Connecting an EX3200 Switch*.
- Performed the initial switch configuration. See *Connecting and Configuring an EX Series Switch (J-Web Procedure)*.

Overview and Topology

EX Series switches connect all devices in an office or data center into a single LAN to provide sharing of common resources such as printers and file servers and to enable wireless devices to connect to the LAN through wireless access points. 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 allows 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, printers, and wireless access points. 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 for this example consists of one EX4200-48P switch, which has a total of 48 Gigabit Ethernet ports, all of which support Power over Ethernet (PoE). Most of the switch ports connect to Avaya IP telephones. The remainder of the ports connect to wireless access points, file servers, and printers. Table 40 on page 154 explains the components of the example topology.

Table 40: Components of the Multiple VLAN Topology

Property	Settings
Switch hardware	EX4200-48P, 48 Gigabit Ethernet ports, all PoE-enabled (**ge-0/0/0** through **ge-0/0/47**)

Table 40: 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**	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**
Interfaces in VLAN **support**	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	**ge-0/0/2** and **ge-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

Configure Layer 2 switching for two VLANs:

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 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 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 wireless access point in the sales VLAN:

```
[edit interfaces ge-0/0/0 unit 0]
user@switch# set description “Sales wireless access point port”
user@switch# set family ethernet-switching vlan members sales
```

2. Configure the interface for the Avaya IP phone in the sales VLAN:

```
[edit interfaces ge-0/0/3 unit 0]
user@switch# set description “Sales phone port”
user@switch# set family ethernet-switching vlan members sales
```

3. Configure the interface for the printer in the sales VLAN:

```
[edit interfaces ge-0/0/22 unit 0]
user@switch# set description “Sales printer port”
user@switch# set family ethernet-switching vlan members sales
```

4. Configure the interface for the file server in the sales VLAN:

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

5. Configure the interface for the wireless access point in the support VLAN:

```
[edit interfaces ge-0/0/24 unit 0]
user@switch# set description “Support wireless access point port”
user@switch# set family ethernet-switching vlan members support
```

6. Configure the interface for the Avaya IP phone in the support VLAN:

```
[edit interfaces ge-0/0/26 unit 0]
user@switch# set description “Support phone port”
user@switch# set family ethernet-switching vlan members support
```

7. Configure the interface for the printer in the support VLAN:

```
[edit interfaces ge-0/0/44 unit 0]
user@switch# set description “Support printer port”
user@switch# set family ethernet-switching vlan members support
```

8. Configure the interface for the file server in the support VLAN:

```
[edit interfaces ge-0/0/46 unit 0]
user@switch# set description “Support file server port”
```

```
user@switch# set family ethernet-switching vlan members support
```

9. Create the subnet for the sales broadcast domain:

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

10. Create the subnet for the support broadcast domain:

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

11. 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
```

12. 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
user@switch# set support l3-interface vlan.1
```

Results Display the results of the configuration:

```
user@switch> show configuration
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-0/0/22 {
        unit 0 {
            description “Sales printer 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-0/0/24 {
        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 support;
          }
        }
      }
      ge-0/0/46 {
        unit 0 {
          description “Support file server port”;
          family ethernet-switching {
            vlan members support;
          }
        }
        vlans {
          unit 0 {
            family inet address 192.0.2.0/25;
          }
          unit 1 {
            family inet address 192.0.2.128/25;
          }
        }
      }
    }
    vlans {
      sales {
        vlan-id 100;
        interface ge-0/0/0.0:
        interface ge-0/0/3/0;
        interface ge-0/0/20.0;
        interface ge-0/0/22.0;
        l3-interface vlan 0;
```

```
    }
    support {
      vlan-id 200;
      interface ge-0/0/24.0:
      interface ge-0/0/26.0;
      interface ge-0/0/44.0;
      interface ge-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

To 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 to the Correct Interfaces on page 159
- Verifying That Traffic Is Being Routed Between the Two VLANs on page 160
- Verifying That Traffic Is Being Switched Between the Two VLANs on page 160

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

Purpose Verify that the VLANs **sales** and **support** 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:

Use the operational mode commands:

```
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-0/1/0.0*, 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
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 **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 **ge-0/0/0.0**, **ge-0/0/3.0**, **ge-0/0/20.0**, and **ge-0/0/22.0**. VLAN **support** has a tag ID of 200 and is associated with interfaces **ge-0/0/24.0**, **ge-0/0/26.0**, **ge-0/0/44.0**, and **ge-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's 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          - ge-0/0/10.0
  default           00:00:5e:00:01:09 Learn          - ge-0/0/13.0
  default           00:19:e2:50:63:e0 Learn          - ge-0/0/23.0
  sales             *                 Flood          - All-members
  sales             00:00:5e:00:07:09 Learn          - ge-0/0/0.0
```

```
support          *                 Flood          - All-members
support          00:00:5e:00:01:01 Learn          - ge-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 **ge-0/0/0.0** and **ge-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 for an EX Series Switch on page 122
- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182
- Understanding Bridging and VLANs on Switches on page 84

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 41 on page 162 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 41: Components of the Topology for Connecting an Access Switch to a Distribution Switch

Property	Settings
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 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 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 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 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 on page 168
- Verifying the VLAN Members and Interfaces on the Distribution Switch on page 169

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 Switches on page 104
- Example: Setting Up Bridging with Multiple VLANs on page 141

Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support

NOTE: This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

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 access switches to a distribution switch:

Requirements

This example uses the following hardware and software components:

- Three EX Series access switches.
- One EX Series distribution switch.

NOTE: In an access switch-distribution switch topology, you can connect EX Series switches that run a version of Junos OS that supports ELS with EX Series switches that do not run a version of Junos OS that supports ELS. However, this example uses switches running ELS only to show how to configure this topology using the ELS CLI.

- Junos OS Release 12.3R2 or later that supports ELS for EX Series switches.

Before you connect an access switch to a distribution switch, be sure you have:

- Installed the switches. See the installation instructions for your switch.
- Performed the initial software configuration on both switches. For information about the initial software configuration for all EX Series switches except the EX9200 Series switches, see *Connecting and Configuring an EX Series Switch (CLI Procedure)*. For information about the initial software configuration for the EX9200 Series switches, see *Connecting and Configuring an EX9200 Switch (CLI Procedure)*.

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 three access switches 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 one of the access switch's uplink modules connects to the distribution switch, to one 1-Gigabit Ethernet port on the distribution switch.

Figure 1 on page 172 shows an EX9200 distribution switch that is connected to three EX4300 access switches.

Figure 1: Sample Access Switch-Distribution Switch Topology

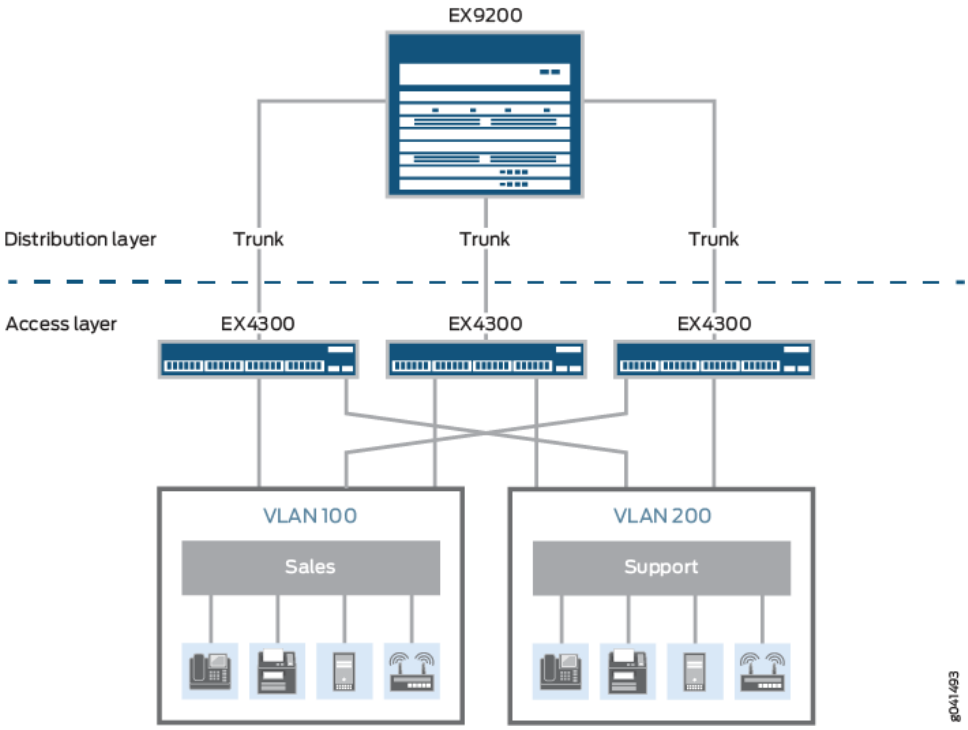


Table 41 on page 162 describes the components of the example topology. The example shows how to configure one of the three access switches. The other access switches could be configured in the same manner.

Table 42: Components of the Topology for Connecting an Access Switch to a Distribution Switch

Property	Settings
Access switch hardware	Three EX4300 switches, each with an uplink module with 1-Gigabit Ethernet ports..
Distribution switch hardware	One EX9208 with up to three EX9200-40T line cards installed, which at full duplex, can provide up to 240 1-Gigabit ports.
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)
Trunk port interfaces	On the access switch: ge-0/2/0 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

Table 42: Components of the Topology for Connecting an Access Switch to a Distribution Switch (continued)

Property	Settings
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

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-0/2/0 unit 0 description "Uplink module port connection to distribution switch"
set interfaces ge-0/2/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/2/0 native-vlan-id 1
set interfaces ge-0/2/0 unit 0 family ethernet-switching vlan members [sales support]
set interfaces ge-0/2/0 unit 0 family ethernet-switching vlan members 1
set interfaces irb unit 0 family inet address 192.0.2.1/25
set interfaces irb unit 1 family inet address 192.0.2.129/25
set vlans sales description “Sales VLAN”
set vlans sales l3-interface irb.0
set vlans sales vlan-id 100
set vlans support description “Support VLAN”
set vlans support vlan-id 200
set vlans support l3-interface irb.1
```

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]
user@access-switch# set ge-0/2/0 unit 0 description "Uplink module port connection to distribution switch"
user@access-switch# set ge-0/2/0 unit 0 family ethernet-switching interface-mode trunk
```

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

```
[edit interfaces]
user@access-switch# set ge-0/2/0 unit 0 family ethernet-switching vlan members [ sales support ]
```

3. To handle untagged packets that are received on the trunk port, create a native VLAN by configuring a VLAN ID and specifying that the trunk port is a member of the native VLAN:

```
[edit interfaces]
user@access-switch# set ge-0/2/0 native-vlan-id 1
user@access-switch# set ge-0/2/0 unit 0 family ethernet-switching vlan members 1
```

4. Configure the sales VLAN:

```
[edit vlans]
user@access-switch# set sales description "Sales VLAN"
user@access-switch# set sales vlan-id 100
user@access-switch# set sales l3-interface irb.0
```

5. Configure the support VLAN:

```
[edit vlans]
user@access-switch# set support description "Support VLAN"
user@access-switch# set support vlan-id 200
user@access-switch# set support l3-interface irb.1
```

6. Create the subnet for the sales VLAN:

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

7. Create the subnet for the support VLAN:

```
[edit interfaces]
user@access-switch# set irb 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 sales
user@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 sales
user@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 sales
user@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
support
user@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
support
user@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
support
user@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
```

Results Display the results of the configuration:

```
user@access-switch> show configuration
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 support;
                }
            }
        }
    }
    ge-0/0/46 {
        unit 0 {
            description "Support file server port";
            family ethernet-switching {
                vlan {
                    members support;
                }
            }
        }
    }
    ge-0/2/0 {
        native-vlan-id 1;
        unit 0 {
            description "Uplinking module connection to distribution switch";
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 1 sales support ];
                }
            }
        }
    }
    irb {
        unit 0 {
            family inet {
                address 192.0.2.1/25;
            }
        }
        unit 1 {
            family inet {
                address 192.0.2.129/25;
            }
        }
    }
}
vlans {
     sales {
        description "Sales VLAN";
        vlan-id 100;
        l3-interface irb.0;
    }
    support {
        description "Support VLAN";
        vlan-id 200;
        l3-interface irb.1;
    }
}
```

TIP: To quickly configure the access 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 unit 0 description "Connection to access switch"
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members [ sales support ]
set interfaces ge-0/0/0 native-vlan-id 1
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 1
set interfaces irb unit 0 family inet address 192.0.2.2/25
set interfaces irb unit 1 family inet address 192.0.2.130/25
set vlans sales description "Sales VLAN"
set vlans sales vlan-id 100
set vlans sales l3-interface irb.0
set vlans support description "Support VLAN"
set vlans support vlan-id 200
set vlans support l3-interface irb.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]
   user@distribution-switch# set ge-0/0/0 unit 0 description "Connection to access switch"
   user@distribution-switch# set ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

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

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

3. To handle untagged packets that are received on the trunk port, create a native VLAN by configuring a VLAN ID and specifying that the trunk port is a member of the native VLAN:

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

4. Configure the sales VLAN:

```
[edit vlans]
user@distribution-switch# set sales description "Sales VLAN"
user@distribution-switch# set sales vlan-id 100
user@distribution-switch# set sales l3-interface irb.0
```

The VLAN configuration for the distribution switch includes the **set l3-interface irb.0** command to route traffic between the sales and support VLANs. The VLAN configuration for the access switch does not include this statement because the access switch is not monitoring IP addresses. Instead, the access switch is passing the IP addresses to the distribution switch for interpretation.

5. Configure the support VLAN:

```
[edit vlans]
user@distribution-switch# set support description "Support VLAN"
user@distribution-switch# set support vlan-id 200
user@distribution-switch# set support l3-interface irb.1
```

The VLAN configuration for the distribution switch includes the **set l3-interface irb.1** command to route traffic between the sales and support VLANs. The VLAN configuration for the access switch does not include this statement because the access switch is not monitoring IP addresses. Instead, the access switch is passing the IP addresses to the distribution switch for interpretation.

6. Create the subnet for the sales VLAN:

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

7. Create the subnet for the support VLAN:

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

Results Display the results of the configuration:

```
user@distribution-switch> show configuration
interfaces {
    ge-0/0/0 {
        native-vlan-id 1;
        unit 0 {
            description "Connection to access switch";
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members [ 1 sales support ];
                }
            }
        }
    }
    irb {
        unit 0 {
            family inet {
                address 192.0.2.2/25;
            }
        }
        unit 1 {
            family inet {
                address 192.0.2.130/25;
            }
        }
    }
}
vlans {
    sales {
        description "Sales VLAN";
        vlan-id 100;
        l3-interface irb.0;
    }
    support {
        description "Support VLAN";
        vlan-id 200;
        l3-interface irb.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 on page 181
- Verifying the VLAN Members and Interfaces on the Distribution Switch on page 181

Verifying the VLAN Members and Interfaces on the Access Switch

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

Action List all VLANs configured on the switch:

```
user@access-switch> show vlans
Routing instance        VLAN name             Tag         Interfaces
default-switch          sales                  100
                                                           ge-0/0/20.0
                                                           ge-0/0/22.0
                                                           ge-0/0/3.0*
                                                           ge-0/0/0.0*
                                                           ge-0/2/0.0*
default-switch          support                200
                                                           ge-0/0/24.0
                                                           ge-0/0/26.0
                                                           ge-0/0/44.0*
                                                           ge-0/0/46.0*
                                                           ge-0/2/0.0*
```

Meaning The output shows the **sales** and **support** VLANs and the interfaces that are configured as members of the respective VLANs.

Verifying the VLAN Members and Interfaces on the Distribution Switch

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

Action List all VLANs configured on the switch:

```
user@distribution-switch> show vlans

Routing instance        VLAN name             Tag         Interfaces
default-switch          sales                  100
                                                           ge-0/0/0.0*
default-switch          support                200
                                                           ge-0/0/0.0*
```

Meaning The output shows the **sales** and **support** VLANs and the interface (ge-0/0/0.0) that is configured as a member of both VLANs. Interface ge-0/0/0.0 is also the trunk interface connected to the access switch.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support on page 131
- Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure) on page 102
- Understanding Bridging and VLANs on Switches on page 84

Example: Connecting an EX Series 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 EX4200-24F 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 EX3200-24P, 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 9.0 or later for EX Series switches

Before you connect an access switch to a distribution switch, be sure you have:

- Installed the two switches. See the installation instructions for your switch.
- Performed the initial software configuration on both switches. See *Connecting and Configuring an EX Series Switch (J-Web Procedure)*.

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.

Figure 2 on page 183 shows one EX4200 switch that is connected to the three access switches.

Figure 2: Topology for Configuration

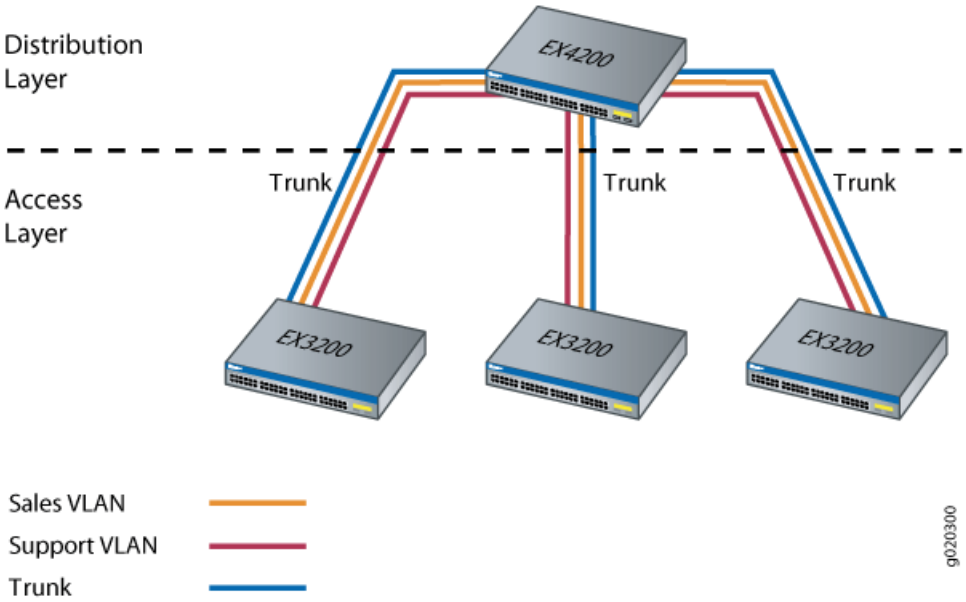


Table 41 on page 162 explains the components of the example topology. The example shows how to configure one of the three access switches. The other access switches could be configured in the same manner.

Table 43: Components of the Topology for Connecting an Access Switch to a Distribution Switch

Property	Settings
Access switch hardware	EX3200-24P, 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	EX4200-24F, 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 **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**)
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**

Table 43: Components of the Topology for Connecting an Access Switch to a Distribution Switch (continued)

Property	Settings
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 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
```

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# set description "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 vlan members [ 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-id 100
user@access-switch# set l3-interface vlan.0
```

5. Configure the support VLAN:

```
[edit vlans support]
user@access-switch# set vlan-id 200
user@access-switch# set l3-interface 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 sales
user@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 sales
user@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 sales
user@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
support
user@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
support
user@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
support
user@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 100
user@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.0
user@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 vlan members [ 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 100
user@distribution-switch# set l3-interface vlan.0
```

   The reason that the VLAN configuration for this distribution switch includes the statement **set l3-interface vlan.0** is that the VLAN is being configured for an attached router. The access switch VLAN configuration did not include this statement because the access switch is not monitoring IP addresses, but is instead passing them to the distribution switch for interpretation.

5. Configure the support VLAN:

```
[edit vlans support]
user@distribution-switch# set vlan-description "Support VLAN"
user@distribution-switch# set vlan-id 200
user@distribution-switch# set l3-interface vlan.1
```

   The reason that the VLAN configuration for this distribution switch includes the statement **set l3-interface vlan.1** is that the VLAN is being configured for an attached router. The access switch VLAN configuration did not include this statement because the access switch is not monitoring IP addresses, but is instead passing them to the distribution switch for interpretation.

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 on page 190
- Verifying the VLAN Members and Interfaces on the Distribution Switch on page 191

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 for an EX Series Switch on page 122
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Understanding Bridging and VLANs on Switches on page 84

Enabling VLAN Pruning for Broadcast, Multicast, and Unknown Unicast Traffic in an EX Series Virtual Chassis (CLI Procedure)

You can enable VLAN pruning for VLANs assigned to interfaces in an EX Series Virtual Chassis. When you enable VLAN pruning for a VLAN in a Virtual Chassis, all broadcast, multicast, and unknown unicast traffic entering that VLAN uses the shortest possible path through the Virtual Chassis to the egress VLAN interface. Enabling VLAN pruning allows you to conserve bandwidth within the Virtual Chassis, since all broadcast, multicast, and unknown unicast traffic in a VLAN is broadcast to all Virtual Chassis member switches when VLAN pruning is disabled.

BEST PRACTICE: We recommend enabling VLAN pruning when configuring a VLAN on an EX Series Virtual Chassis.

To enable VLAN pruning when configuring a VLAN:

```
[edit]
user@switch# set vlans vlan-name vlan-prune
```

Related Documentation

- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98

Configuring a Logical Interface for Access Mode

Enterprise network administrators can configure a single logical interface to accept untagged packets and forward the packets within a specified VLAN. A logical interface configured to accept untagged packets is called an *access interface* or *access port*.

interface-mode access;

You can include this statement at the following hierarchy levels:

- **[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]**
- **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]**

When an untagged or tagged packet is received on an access interface, the packet is accepted, the VLAN ID is added to the packet, and the packet is forwarded within the VLAN that is configured with the matching VLAN ID.

The following example configures a logical interface as an access port with a VLAN ID of 20 on routers and switches that support the enhanced Layer 2 software:

```
[edit interfaces ge-1/2/0]
unit 1 {
  family ethernet-switching {
    interface-mode access;
    vlan members 20;
  }
}
```

Related Documentation

- *802.1Q VLANs Overview*
- *Ethernet Interfaces Feature Guide for Routing Devices*

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 891

Configuring the Native VLAN Identifier (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches and 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 on Switches With ELS Support (CLI Procedure)" on page 196. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

EX Series switches support receiving and forwarding routed or bridged Ethernet frames with 802.1Q VLAN tags. The logical interface on which untagged packets are to be received must be configured with the same native VLAN ID as that configured on the physical interface.

To configure the native VLAN ID using the CLI:

1. Configure the port mode so that the interface is in multiple VLANs and can multiplex traffic between different VLANs. Trunk interfaces typically connect to other switches and to routers on the LAN. Configure the port mode as **trunk**:

   ```
   [edit interfaces ge-0/0/3 unit 0 family ethernet-switching]
   user@switch# set port-mode trunk
   ```

2. Configure the native VLAN ID:

   ```
   [edit interfaces ge-0/0/3 unit 0 family ethernet-switching]
   user@switch# set native-vlan-id 3
   ```

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122

Configuring the Native VLAN Identifier on Switches With ELS Support (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches and Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring the Native VLAN Identifier (CLI Procedure)" on page 195. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

Switches can receive and forward routed or bridged Ethernet frames with 802.1Q VLAN tags. Typically, trunk ports, which connect switches to each other, accept untagged control packets but do not accept untagged data packets. You can enable a trunk port to accept untagged data packets by configuring a native VLAN ID on the interface on which you want the untagged data packets to be received. The logical interface on which untagged packets are to be received must be configured with the same VLAN ID as the native VLAN ID configured on the physical interface.

To configure the native VLAN ID by using the command-line interface (CLI):

1. On the interface on which you want untagged data packets to be received, set the interface mode to **trunk**, which specifies that the interface is in multiple VLANs and can multiplex traffic between different VLANs.:

   ```
   [edit interfaces]
   user@switch# set interface-name unit logical-unit-number family
   ethernet-switching interface-mode trunk
   ```

2. Configure the native VLAN ID:

   ```
   [edit interfaces]
   user@switch# set interface-name native-vlan-id vlan-id
   ```

3. Specify that the logical interface that will receive the untagged data packets is a member of the native VLAN:

   ```
   [edit interfaces]
   user@switch# set interface-name unit logical-unit-number family
   ethernet-switching vlan members vlan-id
   ```

Related Documentation

-
- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support on page 170
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support on page 131
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104

CHAPTER 9

Configuring Learning and Forwarding for VLANs

Layer 2 Learning and Forwarding for VLANs Overview

Understanding Layer 2 Forwarding Tables on Switches, Routers and NFX Series Devices

You can configure Layer 2 MAC address and VLAN learning and forwarding properties in support of Layer 2 bridging. Unicast media access control (MAC) addresses are learned to avoid flooding the packets to all the ports in a VLAN. A source MAC entry is created in its source and destination MAC tables for each MAC address learned from packets received on ports that belong to the VLAN.

When you configure a VLAN, Layer 2 address learning is enabled by default. The VLAN learns unicast media access control (MAC) addresses to avoid flooding the packets to all the ports in the VLAN. Each VLAN creates a source MAC entry in its source and destination MAC tables for each source MAC address learned from packets received on the ports that belong to the VLAN.

NOTE: **Traffic is not flooded back onto the interface on which it was received. However, because this "split horizon" occurs at a late stage, the packet statistics displayed by commands such as show interfaces queue will include flood traffic.**

You can optionally disable MAC learning either for the entire device or for a specific VLAN or logical interface. You can also configure the following Layer 2 learning and forwarding properties:

- Timeout interval for MAC entries
- Static MAC entries for logical interfaces only

- Limit to the number of MAC addresses learned from a specific logical interface or from all the logical interfaces in a VLAN
- Size of the MAC address table for the VLAN
- MAC accounting for a VLAN

For more information about how to configure VLANs and virtual switches, see "Configuring a VLAN" on page 202 and "Configuring a Layer 2 Virtual Switch on an EX Series Switch" on page 95.

Understanding Layer 2 Forwarding Tables on Security Devices

The SRX Series device maintains forwarding tables that contain MAC addresses and associated interfaces for each Layer 2 VLAN. When a packet arrives with a new source MAC address in its frame header, the device adds the MAC address to its forwarding table and tracks the interface at which the packet arrived. The table also contains the corresponding interface through which the device can forward traffic for a particular MAC address.

If the destination MAC address of a packet is unknown to the device (that is, the destination MAC address in the packet does not have an entry in the forwarding table), the device duplicates the packet and floods it on all interfaces in the VLAN other than the interface on which the packet arrived. This is known as *packet flooding* and is the default behavior for the device to determine the outgoing interface for an unknown destination MAC address. Packet flooding is performed at two levels: packets are flooded to different zones as permitted by configured Layer 2 security policies, and packets are also flooded to different interfaces with the same VLAN identifier within the same zone. The device learns the forwarding interface for the MAC address when a reply with that MAC address arrives at one of its interfaces.

You can specify that the SRX Series device use ARP queries and traceroute requests (which are ICMP echo requests with the time-to-live values set to 1) instead of packet flooding to locate an unknown destination MAC address. This method is considered more secure than packet flooding because the device floods ARP queries and traceroute packets—not the initial packet—on all interfaces. When ARP or traceroute flooding is used, the original packet is dropped. The device broadcasts an ARP or ICMP query to all other devices on the same subnetwork, requesting the device at the specified destination IP address to send back a reply. Only the device with the specified IP address replies, which provides the requestor with the MAC address of the responder.

ARP allows the device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnetwork as the ingress IP address. (The ingress IP address refers to the IP address of the last device to send the packet to the device. The device might be the source that sent the packet or a router forwarding the packet.) Traceroute allows the device to discover the destination MAC address even if the destination IP address belongs to a device in a subnetwork beyond that of the ingress IP address.

When you enable ARP queries to locate an unknown destination MAC address, traceroute requests are also enabled. You can also optionally specify that traceroute requests not be used; however, the device can then discover destination MAC addresses for unicast

packets only if the destination IP address is in the same subnetwork as the ingress IP address.

Whether you enable ARP queries and traceroute requests or ARP-only queries to locate unknown destination MAC addresses, the SRX Series device performs the following series of actions:

1. The device notes the destination MAC address in the initial packet. The device adds the source MAC address and its corresponding interface to its forwarding table, if they are not already there.

2. The device drops the initial packet.

3. The device generates an ARP query packet and optionally a traceroute packet and floods those packets out all interfaces except the interface on which the initial packet arrived.

   ARP packets are sent out with the following field values:

   - Source IP address set to the IP address of the IRB
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the MAC address of the IRB
   - Destination MAC address set to the broadcast MAC address (all **0xf**)

   Traceroute (ICMP echo request or ping) packets are sent out with the following field values:

   - Source IP address set to the IP address of the original packet
   - Destination IP address set to the destination IP address of the original packet
   - Source MAC address set to the source MAC address of the original packet
   - Destination MAC address set to the destination MAC address of the original packet
   - Time-to-live (TTL) set to **1**

4. Combining the destination MAC address from the initial packet with the interface leading to that MAC address, the device adds a new entry to its forwarding table.

5. The device forwards all subsequent packets it receives for the destination MAC address out the correct interface to the destination.

Related Documentation

- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- Layer 2 Transparent Mode Overview on page 377
- Understanding Integrated Routing and Bridging on page 445
- Example: Configuring an IRB Interface on a Security Device on page 452
- Example: Configuring the Default Learning for Unknown MAC Addresses on page 57

Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port

Layer 2 learning is enabled by default. A set of VLANs, configured to function as a switch with a Layer 2 trunk port, learns unicast media access control (MAC) addresses to avoid flooding packets to the trunk port.

NOTE: Traffic is not flooded back onto the interface on which it was received. However, because this "split horizon" occurs at a late stage, the packet statistics displayed by commands such as **show interfaces queue** will include flood traffic.

You can optionally disable Layer 2 learning for the entire set of VLANs as well as modify the following Layer 2 learning and forwarding properties:

- Limit the number of MAC addresses learned from the Layer 2 trunk port associated with the set of VLANs
- Modify the size of the MAC address table for the set of VLANs
- Enable MAC accounting for the set of VLANs

Related Documentation

- Layer 2 Learning and Forwarding for VLANs Overview on page 27

Disabling Layer 2 Learning and Forwarding

Disabling dynamic MAC learning on an MX Series router or an EX Series switch prevents all the logical interfaces on the router or switch from learning source and destination MAC addresses.

To disable MAC learning for an MX Series router or an EX Series switch, include the **global-no-mac-learning** statement at the **[edit protocols l2-learning]** hierarchy level:

```
[edit protocols l2-learning]
global-no-mac-learning;
```

For information about how to configure a virtual switch, see *Configuring a Layer 2 Virtual Switch* .

Related Documentation

- *Understanding Layer 2 Learning and Forwarding*
- *Configuring the MAC Table Timeout Interval*
- *Enabling MAC Accounting*
- *Limiting the Number of MAC Addresses Learned from Each Logical Interface*

CHAPTER 10

Configuring 802.1Q VLANs

Layer 2 VLANs Overview

You can configure one or more VLANs to perform Layer 2 bridging. The Layer 2 bridging functions include integrated routing and bridging (IRB) for support for Layer 2 bridging and Layer 3 IP routing on the same interface, and virtual switches that isolate a LAN segment with its spanning-tree protocol instance and separate its VLAN ID space.

A VLAN is a set of logical ports that share the same flooding or broadcast characteristics and span one or more ports of multiple devices.

You can configure one or more VLANs to perform Layer 2 bridging. Thus, MX Series routers or EX Series switches can function as Layer 2 switches, each with multiple bridging, or broadcast, domains that participate in the same Layer 2 network. You can also configure Layer 3 routing support for a VLAN. Integrated routing and bridging (IRB) provides support for Layer 2 bridging and Layer 3 IP routing on the same interface. IRB enables you to route packets to another routed interface or to another VLAN that has a Layer 3 protocol configured.

You can also group one or more VLANs within a single instance, or virtual switch. Multiple virtual switches, each of which operates independently of other virtual switches on the device, are supported. Virtual switches isolate a LAN segment with its spanning-tree protocol instance and separate its VLAN ID space. Thus, each virtual switch can participate in a different Layer 2 network.

VLANs provide support for a Layer 2 trunk port. A Layer 2 trunk interface enables you to configure a single logical interface to represent multiple VLANs on a physical interface. You can configure a set of VLANs and VLAN identifiers that are automatically associated with one or more Layer 2 trunk interfaces. Packets received on a trunk interface are forwarded within a VLAN that has the same VLAN identifier. A Layer 2 trunk interface also supports IRB within a VLAN. In addition, you can configure Layer 2 learning and forwarding properties that apply to the entire set of VLANs.

You can configure VPLS ports in a virtual switch instead of a dedicated routing instance of type **vpls** so that the logical interfaces of the Layer 2 VLANs in the virtual switch can handle VPLS routing instance traffic. Packets received on a Layer 2 trunk interface are forwarded within a VLAN that has the same VLAN identifier.

Configuring a VLAN

A VLAN must include a set of logical interfaces that participate in Layer 2 learning and forwarding. You can optionally configure a VLAN identifier and a Layer 3 interface for the VLAN to also support Layer 3 IP routing.

To enable a VLAN, include the following statements:

```
[edit]
vlans {
  vlan-name {
    domain-type bridge:
    interface interface-name;
    l3-interface interface-name;
    vlan-id (none | all | number);
    vlan-id-list [ vlan-id-numbers ];
    vlan-tags outer number inner number);
  }
}
```

You cannot use the slash (/) character in VLAN names. If you do, the configuration does not commit and an error is generated.

For the **vlan-id** statement, you can specify either a valid VLAN identifier or the **none** or **all** options.

To include one or more logical interfaces in the VLAN, specify an ***interface-name*** for an Ethernet interface you configured at the **[edit interfaces]** hierarchy level.

NOTE: A maximum of 4096 active logical interfaces are supported for a VLAN or on each mesh group in a virtual private LAN service (VPLS) instance configured for Layer 2 bridging.

By default, each VLAN maintains a Layer 2 forwarding database that contains media access control (MAC) addresses learned from packets received on the ports that belong to the VLAN. You can modify Layer 2 forwarding properties, for example, disabling MAC learning for the entire system or a VLAN, adding static MAC addresses for specific logical

interfaces, and limiting the number of MAC addresses learned by the entire system, the VLAN, or a logical interface.

You can also configure spanning tree protocols to prevent forwarding loops.

Configuring VLAN Encapsulation

To configure encapsulation on an interface, enter the **encapsulation** statement at the **[edit interfaces *interface-name*]** hierarchy level:

```
[edit interfaces interface-name]
encapsulation type;
```

The following list contains important notes regarding encapsulation:

- Ethernet interfaces in VLAN mode can have multiple logical interfaces. In CCC and VPLS modes, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for CCC or VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for CCC or VPLS VLANs.
- For encapsulation type **flexible-ethernet-services**, all VLAN IDs are valid.
- For some encapsulation types, including flexible Ethernet services, Ethernet VLAN CCC, and VLAN VPLS, you can also configure the encapsulation type that is used inside the VLAN circuit itself. To do this, include the **encapsulation** statement:

  ```
  encapsulation (vlan-ccc | vlan-tcc | vlan-vpls);
  ```

  You can include this statement at the following hierarchy levels:

  - **[edit interfaces *interface-name* unit *logical-unit-number*]**
  - **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number*]**

- You cannot configure a logical interface with VLAN CCC or VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation or with flexible Ethernet services encapsulation. In general, the logical interface must have a VLAN ID of 512 or higher; if the VLAN ID is 511 or lower, it will be subject to the normal destination filter lookups in addition to source address filtering. However if you configure flexible Ethernet services encapsulation, this VLAN ID restriction is removed.

In general, you configure an interface's encapsulation at the **[edit interfaces *interface-name*]** hierarchy level.

Example: Configuring VLAN Encapsulation on a Gigabit Ethernet Interface

Configure VLAN CCC encapsulation on a Gigabit Ethernet interface:

```
interfaces ge-2/1/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  unit 0 {
    encapsulation vlan-ccc;
    vlan-id 600;
  }
```

```
}
```

Example: Configuring VLAN Encapsulation on an Aggregated Ethernet Interface

Configure VLAN CCC encapsulation on an aggregated Gigabit Ethernet interface:

```
interfaces ae0 {
  vlan-tagging;
  encapsulation vlan-vpls;
  unit 0 {
    vlan-id 100;
  }
}
```

Related Documentation

- *802.1Q VLANs Overview*
- *Ethernet Interfaces Feature Guide for Routing Devices*

Configuring Inner and Outer TPIDs and VLAN IDs

For some rewrite operations, you must configure the inner or outer TPID values and inner or outer VLAN ID values. These values can be applied to either the input VLAN map or the output VLAN map.

On Ethernet IQ, IQ2, and IQ2-E interfaces; on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces; and on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, to configure the inner TPID, include the **inner-tag-protocol-id** statement:

```
inner-tag-protocol-id tpid;
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number* input-vlan-map**]**
- **[edit interfaces** *interface-name* **unit** *logical-unit-number* output-vlan-map**]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number* input-vlan-map**]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number* output-vlan-map**]**

For the inner VLAN ID, include the **inner-vlan-id** statement. For the outer TPID, include the **tag-protocol-id** statement. For the outer VLAN ID, include the **vlan-id** statement:

```
input-vlan-map {
  (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
  inner-tag-protocol-id tpid;
  inner-vlan-id number;
  tag-protocol-id tpid;
  vlan-id number;
}
output-vlan-map {
  (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
```

```
    inner-tag-protocol-id tpid;
    inner-vlan-id number;
    tag-protocol-id tpid;
    vlan-id number;
  }
```

For aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces, include the **tag-protocol-id** statement for the outer TPID. For the outer VLAN ID, include the **vlan-id** statement:

```
input-vlan-map {
    (pop | push | swap);
    tag-protocol-id tpid;
    vlan-id number;
}
output-vlan-map {
    (pop | push | swap);
    tag-protocol-id tpid;
    vlan-id number;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number***]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number***]**

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see *802.1Q VLANs Overview*.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces** *interface-name* **ether-options ethernet-switch-profile tag-protocol-id [** ***tpids*** **]]** hierarchy level.

Table 44 on page 205 and Table 45 on page 206 specify when these statements are required. Table 44 on page 205 indicates valid statement combinations for rewrite operations for the input VLAN map. “No” means the statement must not be included in the input VLAN map for the rewrite operation. “Optional” means the statement may be optionally specified for the rewrite operation in the input VLAN map. “Any” means that you must include the **vlan-id** statement, **tag-protocol-id** statement, **inner-vlan-id** statement, or **inner-tag-protocol-id** statement.

Table 44: Rewrite Operations and Statement Usage for Input VLAN Maps

	Input VLAN Map Statements			
Rewrite Operation	vlan-id	tag-protocol-id	inner-vlan-id	inner-tag-protocol-id
push	Optional	Optional	No	No
pop	No	No	No	No
swap	Any	Any	No	No

Table 44: Rewrite Operations and Statement Usage for Input VLAN Maps (continued)

	Input VLAN Map Statements			
push-push	Optional	Optional	Optional	optional
swap-push	Optional	Optional	Any	Any
swap-swap	Optional	Optional	Any	Any
pop-swap	No	No	Any	Any
pop-pop	No	No	No	No

Table 45 on page 206 indicates valid statement combinations for rewrite operations for the output VLAN map. “No” means the statement must not be included in the output VLAN map for the rewrite operation. “Optional” means the statement may be optionally specified for the rewrite operation in the output VLAN map.

Table 45: Rewrite Operations and Statement Usage for Output VLAN Maps

	Output VLAN Map Statements			
Rewrite Operation	vlan-id	tag-protocol-id	inner-vlan-id	inner-tag-protocol-id
push	No	Optional	No	No
pop	No	No	No	No
swap	No	Optional	No	No
push-push	No	Optional	No	Optional
swap-push	No	Optional	No	Optional
swap-swap	No	Optional	No	Optional
pop-swap	No	No	No	Optional
pop-pop	No	No	No	No

The following examples use Table 44 on page 205 and Table 45 on page 206 and show how the **pop-swap** operation can be configured in an input VLAN map and an output VLAN map:

Input VLAN Map with inner-vlan-id Statement, Output VLAN Map with Optional

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
   pop-swap;
   inner-vlan-id number;
}
output-vlan-map {
```

inner-tag-protocol-id Statement

```
    pop-swap;
    inner-tag-protocol-id tpid;
}
```

Input VLAN Map with inner-tag-protocol-id Statement, Output VLAN Map with Optional inner-tag-protocol-id Statement

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
output-vlan-map {
    pop-swap;
    inner-tag-protocol-id tpid;
}
```

Input VLAN Map with inner-tag-protocol-id and inner-vlan-id Statements

```
[edit interfaces interface-name unit logical-unit-number]
input-vlan-map {
    pop-swap;
    inner-vlan-id number;
    inner-tag-protocol-id tpid;
}
```

Stacking a VLAN Tag

To stack a VLAN tag on all tagged frames entering or exiting the interface, include the **push**, **vlan-id**, and **tag-protocol-id** statements in the input VLAN map or the output VLAN map:

```
input-vlan-map input-vlan-map {
  push;
  vlan-id number;
  tag-protocol-id tpid;
}
output-vlan-map {
  push;
  tag-protocol-id tpid;
}
```

You can include these statements at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number***]**
- **[edit interfaces** *interface-name* **unit** *logical-unit-number***]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number***]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number***]**

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces** *interface-name* **ether-options ethernet-switch-profile tag-protocol-id [** *tpids* **]]** hierarchy level.

Rewriting a VLAN Tag and Adding a New Tag

On Ethernet IQ, IQ2 and IQ2-E interfaces, on MX Series router Gigabit Ethernet, Tri-Rate Ethernet copper, and 10-Gigabit Ethernet interfaces, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and on Gigabit Ethernet and 10-Gigabit Ethernet interfaces on EX Series switches, to replace the outer VLAN tag of the incoming frame with a user-specified VLAN tag value, include the **swap-push** statement in the input VLAN map or output VLAN map:

```
swap-push
```

A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame.

You can include this statement at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number* input-vlan-map**]**
- **[edit interfaces** *interface-name* **unit** *logical-unit-number* output-vlan-map**]**

- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map]
- [edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

See *Rewrite Operations and Statement Usage for Input VLAN Maps* and *Rewrite Operations and Statement Usage for Output VLAN Maps* for information about configuring inner and outer VLAN ID values and inner and outer TPID values required for VLAN maps.

Related Documentation

- input-vlan-map on page 984
- output-vlan-map on page 1058
- swap-push on page 1109
- *unit*
- *Ethernet Interfaces Feature Guide for Routing Devices*

Configuring VLAN Translation with a VLAN ID List

In many cases, the VLAN identifiers on the frames of an interface's packets are not correct. VLAN translation, or VLAN rewrite, allows you to configure bidirectional VLAN identifier translation with a list on frames arriving on and leaving from a logical interface. This lets you use unique VLAN identifiers internally and maintain legacy VLAN identifiers on logical interfaces.

To perform VLAN translation on the packets on a trunk interface, insert the **vlan-rewrite** statement at the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level. You must also include the **interface-mode trunk** statement within the **[edit interfaces *interface-name* unit *unit-number* family ethernet-switching]** hierarchy because VLAN translation is only supported on trunk interfaces. The reverse translation takes place on traffic exiting the interface. In other words, if VLAN 200 is translated to 500 on traffic entering the interface, VLAN 500 is translated to VLAN 200 on traffic leaving the interface.

NOTE: You can configure either flexible VLAN tagging or trunk mode on interfaces. VLAN translation does not support both. Additionally, the **inner-vlan-id-list** statement is supported only on interfaces with VLAN tagging (VLAN IDs).

The following example translates incoming trunk packets from VLAN identifier 200 to 500 and 201 to 501 (other valid VLAN identifiers are not affected):

```
[edit interfaces ge-1/0/1]
unit 0 {
   ... # Other logical interface statements
   family ethernet-switching {
      interface-mode trunk # Translation is only for trunks
      inner-vlan-id-list [ 100 500–600 ];
      vlan-rewrite {
         translate 200 500;
         translate 201 501;
      }
   ... # Other ethernet-switching statements
   }
}
```

NOTE: This example also translates frame VLANs from 500 to 200 and 501 to 201 on egress.

Related Documentation

- Rewriting a VLAN Tag and Adding a New Tag on page 208

Configuring VLAN Translation on Security Devices

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.

Before you begin configuring VLAN translation, make sure you have created and configured the necessary customer VLANs on the neighboring switches. See *Configuring VLANs*.

VLAN translation can be done in two ways:

- To configure VLAN translation in VLAN retagging, an enterprise provider style of VLAN translation can be achieved by following CLI configuration:

  [edit]
  user@host#set interfaces *intf-name* unit 0 family ethernet-switching interface-mode trunk
  user@host#set interfaces *intf-name* unit 0 family ethernet-switching vlan members v1000
  user@host#set interfaces *intf-name* unit 0 family ethernet-switching vlan-rewrite translate 500 1000

- To configure VLAN translation in Q-in-Q, a service provider style of VLAN translation can be achieved by following CLI configuration:

  [edit]
  user@host#set interfaces *intf-name* flexible-vlan-tagging
  user@host#set interfaces *intf-name* encapsulation extended-vlan-bridge
  user@host#set interfaces *intf-name* unit 100 vlan-id 500
  user@host#set interfaces *intf-name* unit 100 input-vlan-map swap
  user@host#set interfaces *intf-name* unit 100 input-vlan-map tag-protocol-id 0x8100
  user@host#set interfaces *intf-name* unit 100 output-vlan-map swap
  user@host#set interfaces *intf-name* unit 100 family ethernet-switching vlan members v1000

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554

Configuring Static MAC Addresses for Logical Interfaces in a VLAN

You can manually add static MAC entries for the logical interfaces in a VLAN. You can specify one or more static MAC addresses for each logical interface.

To add a static MAC address for a logical interface in a VLAN, include the **static-mac *mac-address*** statement at the **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy level.

```
[edit]
vlans {
    vlan-name {
        domain-type bridge;
        switch-options {
```

```
            interface interface-name {
              static-mac mac-address {
                <vlan-id number>;
              }
            }
          }
        }
      }
```

You can optionally specify a VLAN identifier for the static MAC address by using the **vlan-id** statement. To specify a VLAN identifier for a static MAC address, you must use the **all** option when configuring a VLAN identifier for the VLAN.

NOTE: If a static MAC address you configure for a logical interface appears on a different logical interface, packets sent to that interface are dropped.

Related Documentation

- Disabling MAC Learning for a VLAN or Logical Interface on page 64
- Configuring the Size of the MAC Address Table on page 53
- Enabling MAC Accounting for a VLAN on page 69

CHAPTER 11

Configuring Tagged VLANs

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 on Switches with ELS Support" on page 216..

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 219
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Example: Setting Up Bridging with Multiple VLANs on page 141

Creating a Series of Tagged VLANs on Switches with ELS Support

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 Junos OS for QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Creating a Series of Tagged VLANs" on page 214. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

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-id-list [ 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 the same result: VLANs **__employee_120__** through **__employee_130__** are created.

NOTE: **When a series of VLANs is created using the vlan-id-list command, the VLAN names are preceded and followed by a double underscore.**

Related Documentation

- Example: Setting Up Bridging with Multiple VLANs on Switches on page 147
- Understanding Bridging and VLANs on Switches on page 84

Creating a Series of Tagged VLANs on EX Series Switches (CLI Procedure)

To identify which VLAN traffic belongs to, all frames on an Ethernet VLAN are identified by a tag, as defined in the IEEE 802.1Q standard. These frames are *tagged* and are encapsulated with 802.1Q tags. For a simple network that has only a single VLAN, all traffic has the same 802.1Q tag.

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 know 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 **10-12**. 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.
- Voice over IP (VoIP) configurations do not support a range 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 ge-0/0/22.0 family ethernet-switching vlan members employee
```

- Include the VLAN range:

```
[edit interfaces]
user@switch# set interfaces ge-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 have the same result: VLANs **__employee_120__** through **__employee_130__** are created.

NOTE: When a series of VLANs are created using the vlan-range command, the VLAN names are prefixed and suffixed with a double underscore.

Related Documentation

- Verifying That a Series of Tagged VLANs Has Been Created on an EX Series Switch on page 221
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182
- Understanding Bridging and VLANs on Switches on page 84

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 214
- Creating a Series of Tagged VLANs on Switches with ELS Support on page 216

Verifying That a Series of Tagged VLANs Has Been Created on an EX Series Switch

Purpose Verify that a series of tagged VLANs is created on the switch.

Action Display the VLANs in the ascending order of their VLAN ID:

```
user@switch>  show vlans sort-by tag

Name             Tag        Interfaces
__employee_120__  120
                            ge-0/0/22.0*
__employee_121__  121
                            ge-0/0/22.0*
__employee_122__  122
                            ge-0/0/22.0*
__employee_123__  123
                            ge-0/0/22.0*
__employee_124__  124
                            ge-0/0/22.0*
__employee_125__  125
                            ge-0/0/22.0*
__employee_126__  126
                            ge-0/0/22.0*
__employee_127__  127
                            ge-0/0/22.0*
__employee_128__  128
                            ge-0/0/22.0*
__employee_129__  129
                            ge-0/0/22.0*
__employee_130__  130
                            ge-0/0/22.0*
```

Display the VLANs by the alphabetical order of the VLAN name:

```
user@switch>  show vlans sort-by name

Name             Tag   Interfaces

__employee_120__ 120
                       ge-0/0/22.0*
__employee_121__ 121
                       ge-0/0/22.0*
__employee_122__ 122
                       ge-0/0/22.0*
__employee_123__ 123
                       ge-0/0/22.0*
__employee_124__ 124
                       ge-0/0/22.0*
__employee_125__ 125
                       ge-0/0/22.0*
__employee_126__ 126
                       ge-0/0/22.0*
__employee_127__ 127
                       ge-0/0/22.0*
__employee_128__ 128
                       ge-0/0/22.0*
__employee_129__ 129
                       ge-0/0/22.0*
__employee_130__ 130
                       ge-0/0/22.0*
```

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
                       ge-0/0/22.0*
__employee_121__ 121
                       ge-0/0/22.0*
__employee_122__ 122
                       ge-0/0/22.0*
__employee_123__ 123
                       ge-0/0/22.0*
__employee_124__ 124
                       ge-0/0/22.0*
__employee_125__ 125
                       ge-0/0/22.0*
__employee_126__ 126
                       ge-0/0/22.0*
__employee_127__ 127
                       ge-0/0/22.0*
__employee_128__ 128
                       ge-0/0/22.0*
__employee_129__ 129
                       ge-0/0/22.0*
__employee_130__ 130
                       ge-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 **ge-0/0/22.0**. The asterisk (*****) beside 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 prefixed and suffixed with a double underscore.

Related Documentation

- Creating a Series of Tagged VLANs on EX Series Switches (CLI Procedure) on page 218

CHAPTER 12

Configuring Private VLANs

Understanding Private VLANs

VLANs limit broadcasts to specified users. Private VLANs (PVLANs) take this concept a step further by limiting communication within a VLAN. PVLANs accomplish this by restricting traffic flows through their member switch ports (which are 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 or link aggregation group (LAG) 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 port, thereby preventing the ports from communicating with each other.

PVLANs provide Layer 2 isolation between ports within a VLAN, splitting a broadcast domain into multiple discrete broadcast subdomains by creating secondary VLANs (community VLANs and an isolated VLAN) inside a primary VLAN. Ports within the same community VLAN can communicate with each other. Ports within an isolated VLAN can communicate *only* with a single uplink port.

Just like regular VLANs, PVLANs are isolated on Layer 2 and require one of the following options to route Layer 3 traffic among the secondary VLANs:

- A promiscuous port connection with a router
- A routed VLAN interface (RVI)

NOTE: **To route Layer 3 traffic among secondary VLANs, a PVLAN needs only one of the options mentioned above. If you use an RVI, you can still implement a promiscuous port connection to a router with the promiscuous port set up to handle only traffic that enters and exits the PVLAN.**

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 each other. Another typical use for a PVLAN is to provide per-room Internet access in a hotel.

NOTE: **You can configure a PVLAN to span switches that support PVLANs.**

This topic explains the following concepts regarding PVLANs on EX Series switches:

Why Use PVLANs

PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between hosts.The need to segregate a single VLAN is particularly useful in the following deployment scenarios:

- Server farms—A typical Internet service provider uses a server farm to provide Web hosting for numerous customers. Locating the various servers within a single server farm provides ease of management. Security concerns arise if all servers are in the same VLAN because Layer 2 broadcasts go to all servers in the VLAN.
- Metropolitan Ethernet networks—A metro service provider offers Layer 2 Ethernet access to assorted homes, rental communities, and businesses. The traditional solution of deploying one VLAN per customer is not scalable and is difficult to manage, leading to potential waste of IP addresses. PVLANs provide a more secure and more efficient solution.

Typical Structure and Primary Application of PVLANs

A PVLAN can be configured on a single switch or can be configured to span multiple switches. The types of domains and ports are:

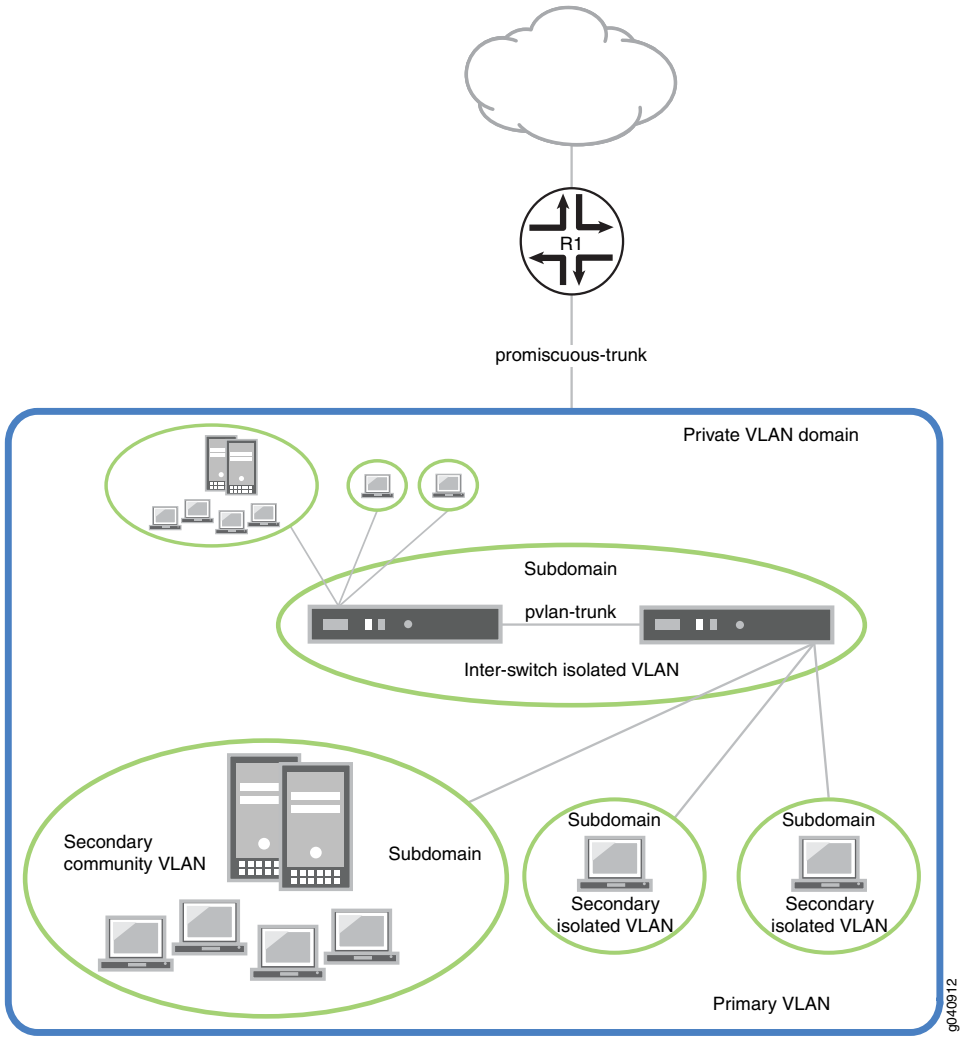
- Primary VLAN—The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. The primary PVLAN can contain multiple secondary VLANs (one isolated VLAN and multiple community VLANs).
- Isolated VLAN/isolated port—A primary VLAN can contain only one isolated VLAN. An interface within an isolated VLAN can forward packets only to a promiscuous port or the Inter-Switch Link (ISL) port. An isolated interface cannot forward packets to another isolated interface; and an isolated interface cannot receive packets from another isolated interface. If a customer device needs to have access *only* to a gateway router, the device must be attached to an isolated trunk port.
- Community VLAN/community port—You can configure multiple community VLANs within a single PVLAN. An interface within a specific community VLAN can establish Layer 2 communications with any other interface that belongs to the same community VLAN. An interface within a community VLAN can also communicate with a promiscuous port or the ISL port. If you have, for example, two customer devices that you need to isolate from other customer devices but that must be able to communicate with one another, use community ports.
- Promiscuous port—A promiscuous port has Layer 2 communications with all interfaces in the PVLAN, regardless of whether an interface belongs to an isolated VLAN or a community VLAN. A promiscuous port is a member of the primary VLAN but is not included within any secondary subdomain. Layer 3 gateways, DHCP servers, and other

trusted devices that need to communicate with endpoint devices are typically connected to a promiscuous port.

- Inter-Switch Link (ISL)—An ISL is a trunk port that connects multiple switches in a PVLAN and contains two or more VLANs. It is required only when a PVLAN spans multiple switches.

The configured PVLAN is the *primary* domain (primary VLAN). Within the PVLAN, you configure *secondary* VLANs, which become subdomains nested within the primary domain. A PVLAN can be configured on a single switch or can be configured to span multiple switches. The PVLAN shown in Figure 3 on page 228 includes two switches, with a primary PVLAN domain and various subdomains.

Figure 3: Subdomains in a PVLAN

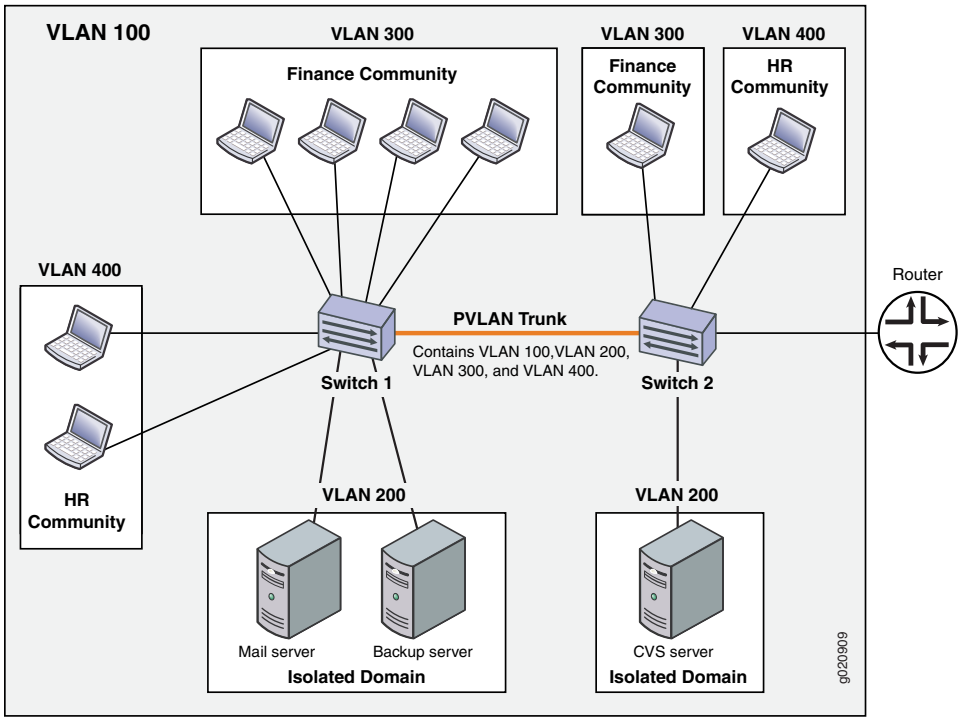


As shown in Figure 5 on page 231, a PVLAN has only one primary domain and multiple secondary domains. The types of domains are:

- Primary VLAN—VLAN used to forward frames downstream to isolated and community VLANs. The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. The primary PVLAN can contain multiple secondary VLANs (one isolated VLAN and multiple community VLANs).
- Secondary isolated VLAN—VLAN that receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN. The isolated VLAN is a secondary VLAN nested within the primary VLAN. A primary VLAN can contain only one isolated VLAN. An interface within an isolated VLAN (isolated interface) can forward packets only to a promiscuous port or the PVLAN trunk port. An isolated interface cannot forward packets to another isolated interface; nor can an isolated interface receive packets from another isolated interface. If a customer device needs to have access *only* to a router, the device must be attached to an isolated trunk port.
- 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. An interswitch isolated VLAN is a secondary VLAN nested within the primary VLAN.
- 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. A community VLAN is a secondary VLAN nested within the primary VLAN. You can configure multiple community VLANs within a single PVLAN. An interface within a specific community VLAN can establish Layer 2 communications with any other interface that belongs to the same community VLAN. An interface within a community VLAN can also communicate with a promiscuous port or the PVLAN trunk port.

Figure 4 on page 230 shows a PVLAN spanning multiple switches, where the primary VLAN (**100**) contains two community domains (**300** and **400**) and one interswitch isolated domain.

Figure 4: PVLAN Spanning Multiple Switches

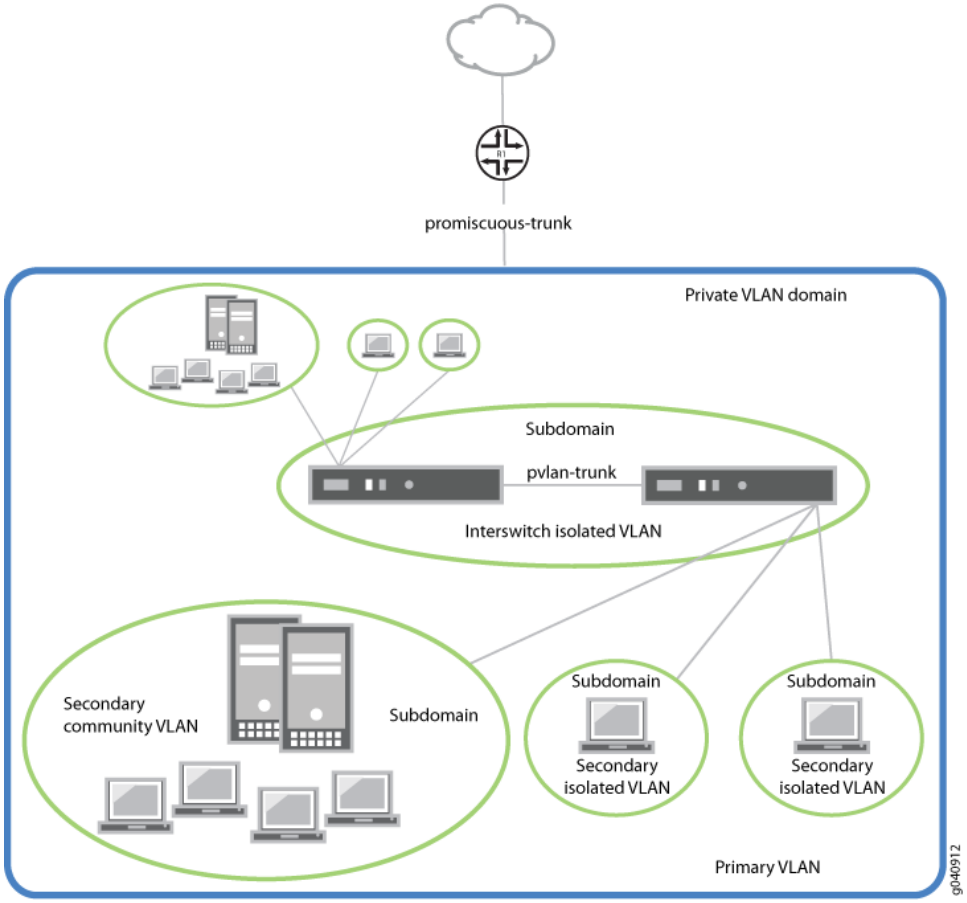


NOTE: Primary and secondary VLANs count against the limit of 4089 VLANs supported on the QFX Series. For example, each VLAN in Figure 4 on page 230 counts against this limit.

Typical Structure and Primary Application of PVLANs on MX Series Routers

The configured PVLAN becomes the primary domain, and secondary VLANs become subdomains that are nested inside the primary domain. A PVLAN can be created on a single router. The PVLAN shown in Figure 5 on page 231 includes one router, with one primary PVLAN domain and multiple secondary subdomains.

Figure 5: Subdomains in a PVLAN With One Router



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 router to another through PVLAN trunk ports.
- Secondary community VLAN—VLAN used to transport frames among members of a community, which is a subset of users within the VLAN, and to forward frames upstream to the primary VLAN.

NOTE: PVLANs are supported on MX80 routers, on MX240, MX480, and MX960 routers with DPCs in enhanced LAN mode, on MX Series routers with MPC1, MPC2, and Adaptive Services PICs.

Typical Structure and Primary Application of PVLANs on EX Series Switches

NOTE: The primary VLAN of the PVLAN is defined with an 802.1Q tag (VLAN ID) for the complete PVLAN. On EX9200 switches, each secondary VLAN must also be defined with its own separate VLAN ID.

Figure 6 on page 232 shows a PVLAN on a single switch, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN **50**).

Figure 6: Private VLAN on a Single EX Switch

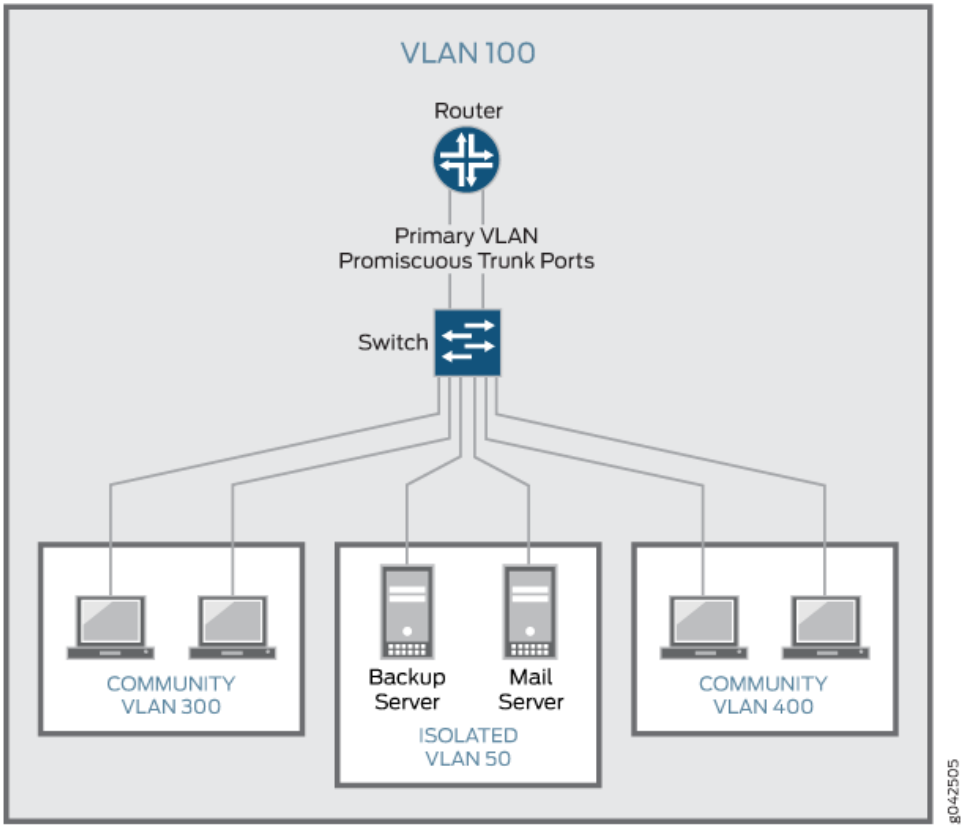
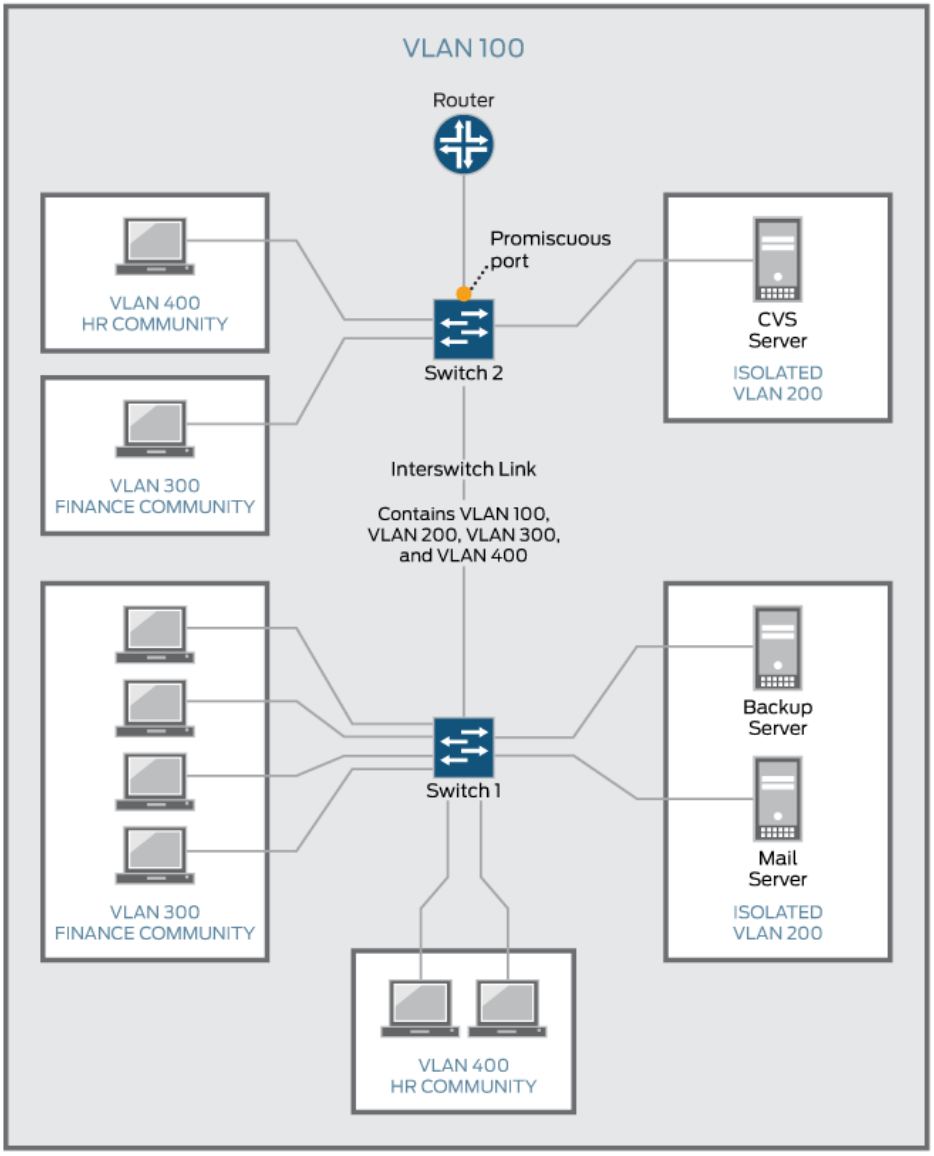


Figure 7 on page 233 shows a PVLAN spanning multiple switches, where the primary VLAN (VLAN **100**) contains two community VLANs (VLAN **300** and VLAN **400**) and one isolated VLAN (VLAN 200). It also shows that Switches 1 and 2 are connected through an interswitch link (PVLAN trunk link).

Figure 7: PVLAN Spanning Multiple EX Series Switches



Also, the PVLANs shown in Figure 6 on page 232 and Figure 7 on page 233 use a promiscuous port connected to a router as the means to route Layer 3 traffic among the community and isolated VLANs. Instead of using the promiscuous port connected to a router, you can configure an RVI on the switch in Figure 6 on page 232 or one of the switches shown in Figure 7 on page 233 (on some EX switches).

To route Layer 3 traffic between isolated and community VLANs, you must either connect a router to a promiscuous port, as shown in Figure 6 on page 232 and Figure 7 on page 233, or configure an RVI.

If you choose the RVI option, you must configure one RVI for the primary VLAN in the PVLAN domain. This RVI serves the entire PVLAN domain regardless of whether the domain includes one or more switches. After you configure the RVI, Layer 3 packets received by the secondary VLAN interfaces are mapped to and routed by the RVI.

When setting up the RVI, you must also enable proxy Address Resolution Protocol (ARP) so that the RVI can handle ARP requests received by the secondary VLAN interfaces.

For information about configuring PVLANs on a single switch and on multiple switches, see "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 271. For information about configuring an RVI, see "Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch (CLI Procedure)" on page 341.

Routing Between Isolated and Community VLANs

To route Layer 3 traffic between isolated and community VLANs, you must connect an external router or switch to a trunk port of the primary VLAN. The trunk port of the primary VLAN is a *promiscuous* port; therefore, it can communicate with *all* the ports in the PVLAN.

PVLANs Use 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 46 on page 234 indicates when a VLAN 802.1Q tag is needed on the primary VLAN or on secondary VLANs.

Table 46: When VLANs in a PVLAN Need 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.

PVLANs Use IP Addresses Efficiently

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 and Forwarding Rules

PVLANs can use up to six different port types. The network depicted in Figure 4 on page 230 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 has Layer 2 communications with all the interfaces that are in the PVLAN, regardless of whether the interface belongs to an isolated VLAN or a community VLAN. A promiscuous port is a member of the primary VLAN, but is not included within one of the secondary subdomains. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with endpoint devices are typically connected to a promiscuous port.
- PVLAN trunk link—The PVLAN trunk link, which is also known as the interswitch link, is required only when a PVLAN is configured to span multiple switches. The PVLAN trunk link connects the multiple switches that compose the 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 other than the isolated 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.
- Interswitch link port—An interswitch link (ISL) port is a trunk port that connects two routers when a PVLAN spans those routers. The ISL port is a member of all VLANs within the PVLAN (that is, the primary VLAN, the community VLANs, and the isolated VLAN).

  Communication between an ISL port and an isolated port is unidirectional. An ISL port's membership in the interswitch isolated VLAN is egress-only, meaning that incoming traffic on the ISL port is never assigned to the isolated VLAN. An isolated port can forward packets to a PVLAN trunk port, but a PVLAN trunk port cannot forward packets to an isolated port. Table 48 on page 236 summarizes whether Layer 2 connectivity exists between the different types of ports.

Table 47 on page 236 summarizes Layer 2 connectivity between the different types of ports within a PVLAN on EX Series switches that support ELS.

Table 47: PVLAN Ports and Layer 2 Forwarding on EX Series switches that support ELS

From Port Type	To Isolated Ports?	To Promiscuous Ports?	To Community Ports?	To Inter-Switch Link Port?
Isolated	Deny	Permit	Deny	Permit
Promiscuous	Permit	Permit	Permit	Permit
Community 1	Deny	Permit	Permit	Permit

Table 48: 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

Table 49 on page 237 summarizes whether or not Layer 2 connectivity exists between the different types of ports within a PVLAN.

Table 49: PVLAN Ports and Layer 2 Connectivity on EX Series Switches without ELS Support

Port Type To: → From:↓	Promiscuous	Community	Isolated	PVLAN Trunk	RVI
Promiscuous	Yes	Yes	Yes	Yes	Yes
Community	Yes	Yes—same community only	No	Yes	Yes
Isolated	Yes	No	No	Yes NOTE: This communication is unidirectional.	Yes
PVLAN trunk	Yes	Yes—same community only	Yes NOTE: This communication is unidirectional.	Yes	Yes
RVI	Yes	Yes	Yes	Yes	Yes

As noted in Table 49 on page 237, Layer 2 communication between an isolated port and a PVLAN trunk port is unidirectional. That is, an isolated port can only send packets to a PVLAN trunk port, and a PVLAN trunk port can only receive packets from an isolated port. Conversely, a PVLAN trunk port cannot send packets to an isolated port, and an isolated port cannot receive packets from a PVLAN trunk port.

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

Creating a PVLAN

The flowchart shown in Figure 8 on page 238 gives you a general idea of the process for creating PVLANs. If you complete your configuration steps in the order shown, you will not violate these PVLAN rules. (In the PVLAN rules, configuring the PVLAN trunk port applies only to a PVLAN that spans multiple routers.)

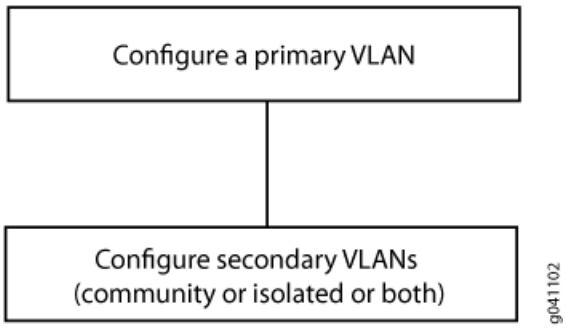
- The primary VLAN must be a tagged VLAN.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN.

- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN.

NOTE: Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Configuring a VLAN on a single router is relatively simple, as shown in Figure 8 on page 238.

Figure 8: Configuring a PVLAN on a Single Switch



Configuring a primary VLAN consists of these steps:

1. Configure the primary VLAN name and 802.1Q tag.
2. Set **no-local-switching** on the primary VLAN.
3. Configure the promiscuous trunk port and access ports.
4. Make the promiscuous trunk and access ports members of the primary VLAN.

Within a primary VLAN, you can configure secondary community VLANs or secondary isolated VLANs or both. Configuring a secondary community VLAN consists of these steps:

1. Configure a VLAN using the usual process.
2. Configure access interfaces for the VLAN.
3. Assign a primary VLAN to the community VLAN,

Isolated VLANs are created internally when the isolated VLAN has access interfaces as members and the option **no-local-switching** is enabled on the primary VLAN.

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.

Trunk ports are only needed for multirouter PVLAN configurations—the trunk port carries traffic from the primary VLAN and all secondary VLANs.

Limitations of Private VLANs

The following constraints apply to private VLAN configurations:

- An access interface can belong to only one PVLAN domain, that is, it cannot participate in two different primary VLANs.
- A trunk interface can be a member of two secondary VLANs as long as the secondary VLANs are in two *different* primary VLANs. A trunk interface cannot be a member of two secondary VLANs that are in the *same* primary VLAN.
- A single region of Multiple Spanning Tree Protocol (MSTP) must be configured on all VLANs that are included within the PVLAN.
- VLAN Spanning Tree Protocol (VSTP) is not supported.
- 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.
- Some configuration statements cannot be specified on a secondary VLAN. You can configure the following statements at the **[edit vlans *vlan-name* switch-options]** hierarchy level *only* on the primary PVLAN.
- If you want to change a primary VLAN to be a secondary VLAN, you must first change it to a normal VLAN and commit the change. For example, you would follow this procedure:

  1. Change the primary VLAN to be a normal VLAN.
  2. Commit the configuration.
  3. Change the normal VLAN to be a secondary VLAN.
  4. Commit the configuration.

  Follow the same sequence of commits if you want to change a secondary VLAN to be a primary VLAN. That is, make the secondary VLAN a normal VLAN and commit that change and then change the normal VLAN to be a primary VLAN.

The following features are *not* supported on PVLANs on Junos switches with support for the ELS configuration style:

- DHCP security features (DHCP snooping, dynamic ARP inspection, IP source guard)
- Egress VLAN firewall filters
- Ethernet ring protection (ERP)
- Flexible VLAN tagging

- global-mac-statistics
- Integrated routing and bridging (IRB) interface
- Multicast snooping or IGMP snooping
- Multichassis link aggregation groups (MC-LAGs)
- Port mirroring
- Q-in-Q tunneling
- VLAN Spanning Tree Protocol (VSTP)
- Voice over IP (VoIP)

You can configure the following statements at the **[edit vlans *vlan-name* switch-options]** hierarchy level only on the primary PVLAN:

- mac-table-size
- no-mac-learning
- mac-statistics
- interface-mac-limit

Related Documentation

- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 244
- Creating a Private VLAN on a Single QFX Switch on page 269

  Example: Configuring a Private VLAN on a Single Switch with ELS Support on page 291
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- Example: Configuring a Private VLAN on a Single EX Series Switch on page 284
- Understanding Bridging and VLANs on Switches on page 84
- Example: Configuring a Private VLAN Spanning Multiple EX Series Switches on page 326
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277
- *Bridge Domains Setup in PVLANs on MX Series Routers*
- *Bridging Functions With PVLANs*

Understanding PVLAN Traffic Flows Across Multiple Switches

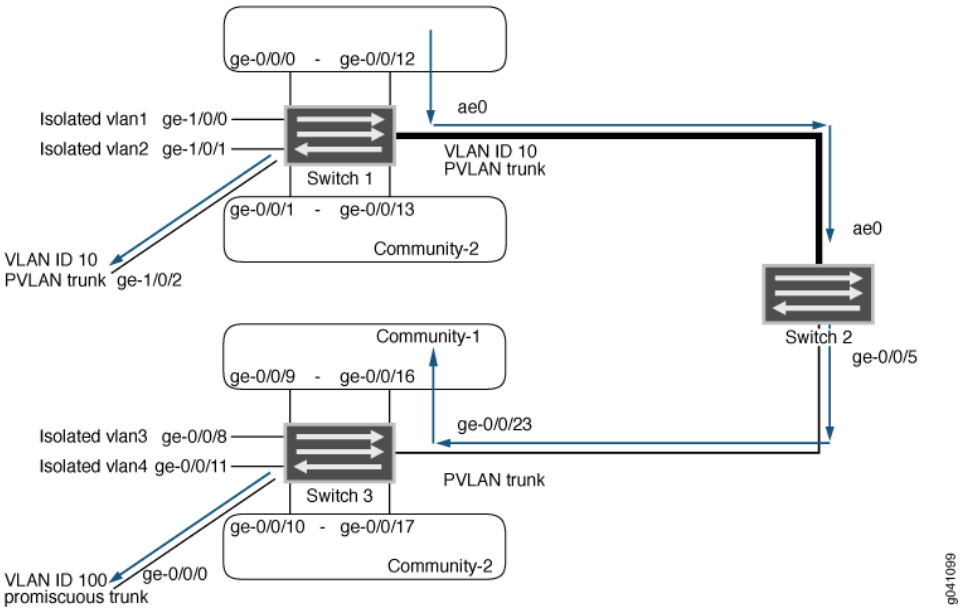
This topic illustrates and explains three different traffic flows on a sample multiswitch network configured with a private VLAN (PVLAN). PVLANs restrict traffic flows through their member switch ports (which are called “private ports”) so that they communicate only with a specific uplink trunk port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

This topic describes:

Community VLAN Sending Untagged Traffic

In this scenario, a VLAN in Community-1 of Switch 1 at interface ge-0/0/0 sends untagged traffic. The arrows in Figure 9 on page 241 represent this traffic flow.

Figure 9: 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

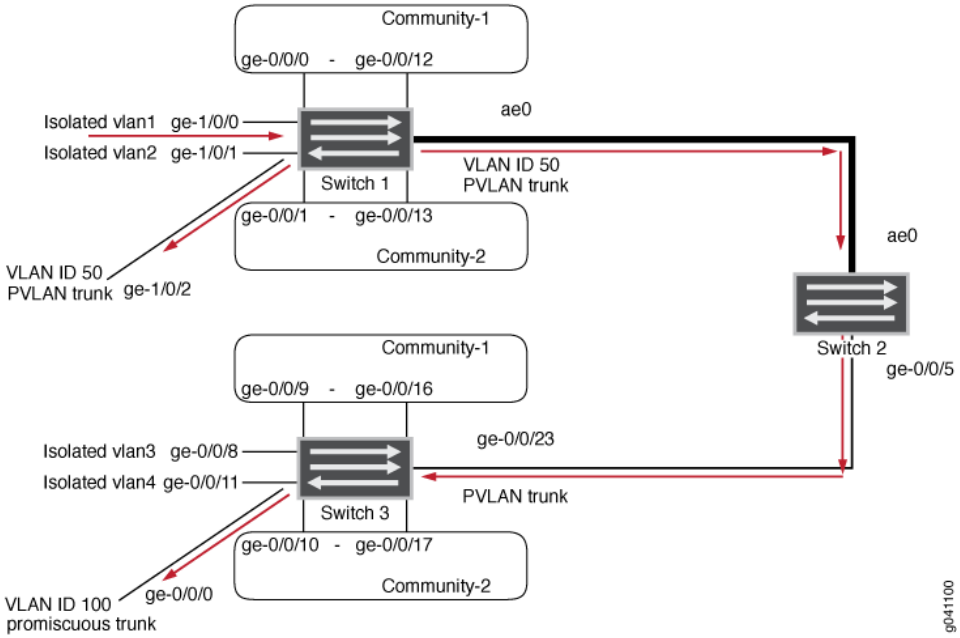
In this scenario, this activity takes place on Switch 3:

- 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 10 on page 242 represent this traffic flow.

Figure 10: 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

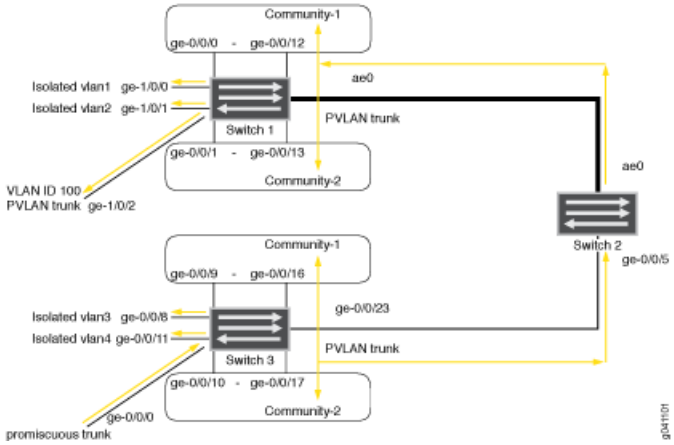
In this scenario, this activity takes place on Switch 3:

- 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 11 on page 243 represent this traffic flow.

Figure 11: 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 page 226

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 244
- Secondary VLAN Trunk Port Details on page 245
- Use Cases on page 246

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

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

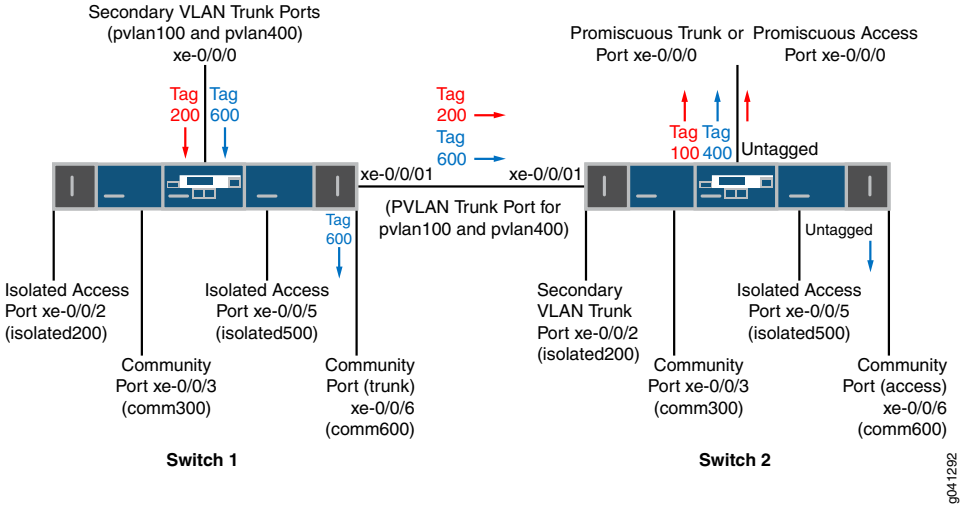
- Primary VLAN pvlan100 with tag 100.
  - Isolated VLAN isolated200 with tag 200 is a member of pvlan100.
  - Community VLAN comm300 with tag 300 is a member of pvlan100.
- Primary VLAN pvlan400 with tag 400.
  - Isolated VLAN isolated500 with tag 500 is a member of pvlan400.
  - Community VLAN comm600 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 12 on page 247 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 12: 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

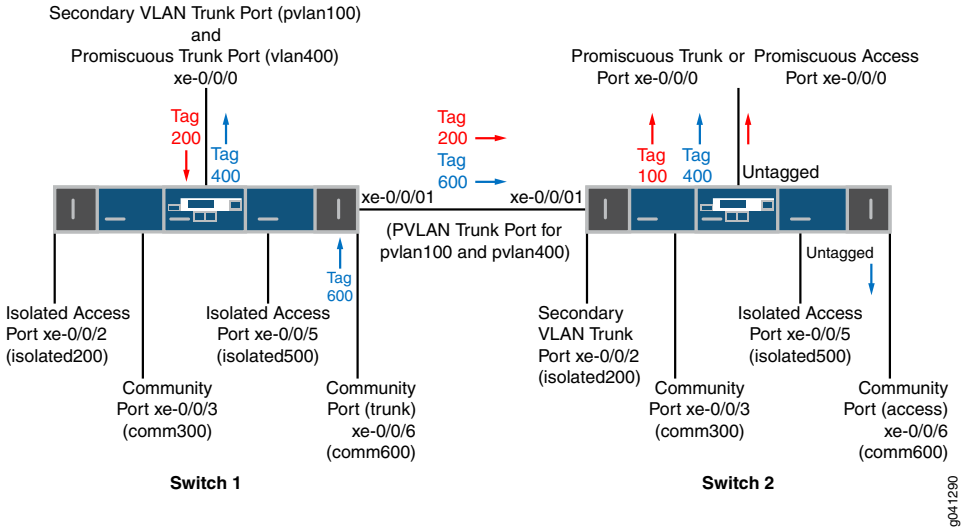
4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2. In this case, the traffic is untagged because the port mode is access.

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 13 on page 249 shows this topology and how traffic for isolated200 (member of pvlan100) and comm600 (member of pvlan400) would flow after ingressing on Switch 1.

Figure 13: 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.

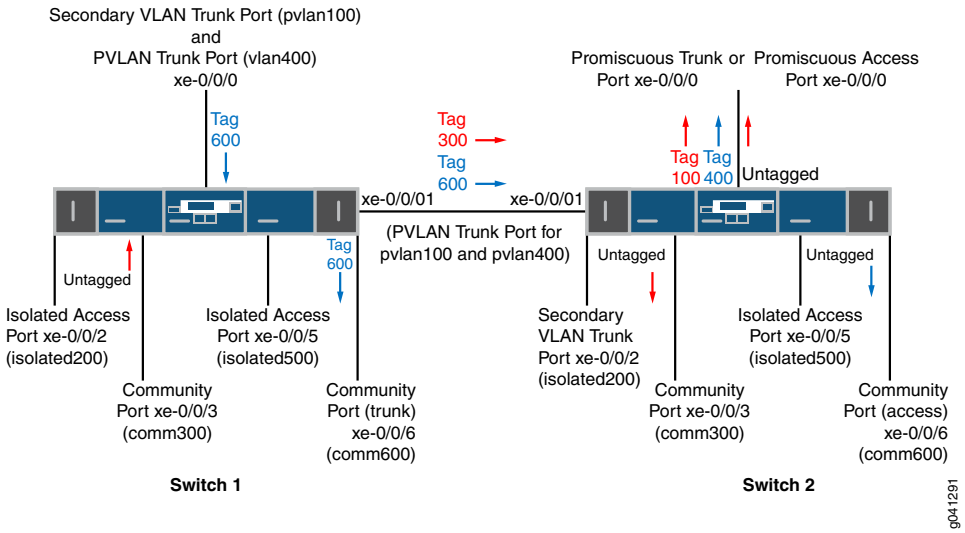
4. Traffic for comm600 also egresses on community port xe-0/0/6 on Switch 2.

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 14 on page 250 shows this topology and how traffic for comm300 (member of pvlan100) and comm600 (member of pvlan400) would flow after ingressing on Switch 1.

Figure 14: 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.

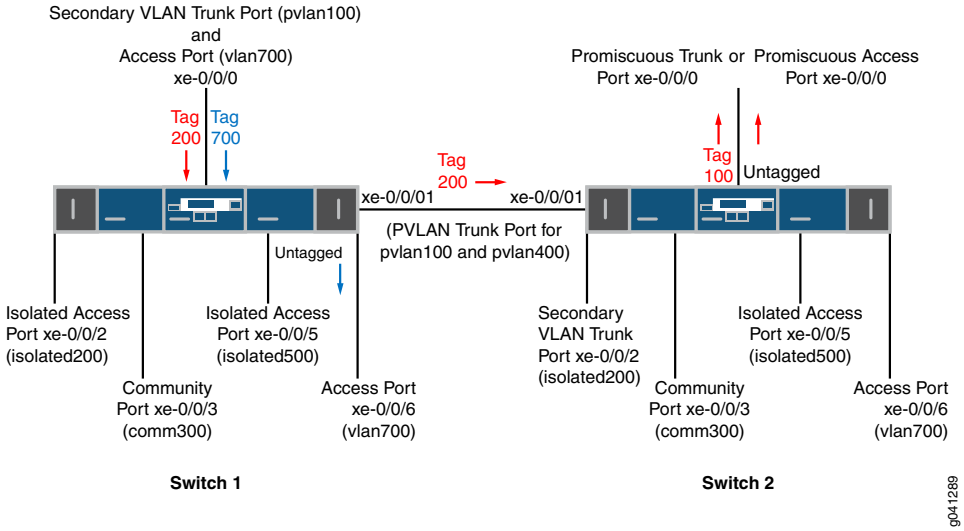
Secondary VLAN Trunk and Non-Private VLAN Interface

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 15 on page 252 shows this topology and how traffic for isolated200 (member of pvlan100) and vlan700 would flow after ingressing on Switch 1.

Figure 15: 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.

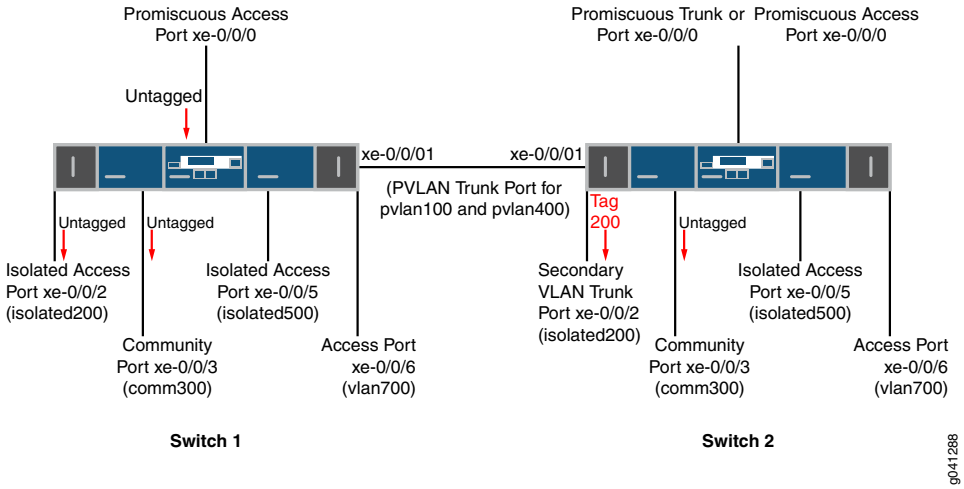
Note that traffic for VLAN isolated200 does not egress on isolated access port xe-0/0/2 on Switch 1 or secondary VLAN trunk port xe-0/0/2 on Switch 2 even though these two ports are members of the same isolated VLAN.

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 16 on page 253 shows this topology and how untagged traffic would flow after ingressing through this interface on Switch 1.

Figure 16: 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 226
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch on page 342
- Creating a Private VLAN on a Single QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- Understanding Egress Firewall Filters with PVLANs on page 253
- Troubleshooting Private VLANs on QFX Switches on page 360

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 226
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch on page 342
- Creating a Private VLAN on a Single QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275

Using 802.1X Authentication and Private VLANs Together on the Same Interface

- Understanding Using 802.1X Authentication and PVLANs Together on the Same Interface on page 254
- Configuration Guidelines for Combining 802.1X Authentication with PVLANs on page 255
- Example: Configuring 802.1X Authentication with Private VLANs in One Configuration on page 255

Understanding Using 802.1X Authentication and PVLANs Together on the Same Interface

You can now configure both 802.1X authentication and private VLANs (PVLANs) on the same interface.

IEEE 802.1X authentication provides network edge security, protecting Ethernet LANs from unauthorized user access by blocking all traffic to and from a supplicant (client) at the interface until the supplicant's credentials are presented and matched on the *authentication server* (a RADIUS server).

Private VLANs (PVLANs) provide Layer 2 isolation between ports within a VLAN, splitting a broadcast domain into multiple discrete broadcast subdomains by creating secondary VLANs. PVLANs are useful for restricting the flow of broadcast and unknown unicast traffic and for limiting the communication between known hosts.

On a switch that is configured with both 802.1X authentication and PVLANs, when a new device is attached to the PVLAN network, the device is authenticated and then is assigned to a secondary VLAN based on the PVLAN configuration or RADIUS profile. The device then obtains an IP address and is given access to the PVLAN network.

NOTE: **This document does not provide detailed information about 802.1X authentication or private VLANs. For those details, see the feature documentation that is specific to those individual features. For 802.1X, see** User Access and Authentication Feature Guide**. For PVLANs, see** Ethernet Switching Feature Guide**.**

Configuration Guidelines for Combining 802.1X Authentication with PVLANs

Keep the following guidelines and limitations in mind for configuring these two features on the same interface:

- You cannot configure an 802.1X-enabled interface as a promiscuous interface (an interface that is a member of the primary VLAN by configuration) or as an interswitch-link (ISL) interface.
- Multiple users cannot be authenticated over different VLANs belonging to the same PVLAN domain on a logical interface—for example, if interface ge-0/0/0 is configured as **supplicant multiple** and clients C1 and C2 are authenticated and are added to dynamic VLANs V1 and V2, respectively, then V1 and V2 must belong to different PVLAN domains.
- If the VoIP VLAN and the data VLAN are different, those two VLANs must be in different PVLAN domains.
- When PVLAN membership is changed (that is, an interface is reconfigured in a different PVLAN), clients must be reauthenticated.

Example: Configuring 802.1X Authentication with Private VLANs in One Configuration

Requirements

- Junos OS Release 18.2R1 or later
- EX2300, EX3400, or EX4300 switch

Before you begin, specify the RADIUS server or servers to be used as the authentication server. See *Specifying RADIUS Server Connections on Switches (CLI Procedure)*.

Overview

The following configuration section shows the access profile configuration, the 802.1X authentication configuration, and finally the VLANs (including PVLANs) configuration.

Configuring 802.1X Authentication with Private VLANs in One Configuration

CLI Quick Configuration

```
[edit]
set access radius-server 10.20.9.199 port 1812
set access radius-server 10.20.9.199 secret "$9$Lqa7dsaZjP5F245Fn/0OX7-V24JGDkmf"
set access profile dot1x-auth authentication-order radius
set access profile authp authentication-order radius
set access profile authp radius authentication-server 10.204.96.165
set switch-options voip interface ge-0/0/8.0 vlan voip
set interfaces ge-0/0/8 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/8 unit 0 family ethernet-switching vlan members data
set protocols dot1x authenticator authentication-profile-name authp
set protocols dot1x authenticator interface ge-0/0/8.0 supplicant multiple
set protocols dot1x authenticator interface ge-0/0/8.0 mac-radius
set vlans community vlan-id 20
set vlans community private-vlan community
set vlans community-one vlan-id 30
set vlans community-one private-vlan community
set vlans isolated vlan-id 200
set vlans isolated private-vlan isolated
set vlans pvlan vlan-id 2000
set vlans pvlan isolated-vlan isolated
set vlans pvlan community-vlans [community community-one]
set vlans data vlan-id 43
set vlans voip vlan-id 33
```

Step-by-Step Procedure

To configure 802.1X authentication and PVLANs in one configuration:

1. Configure the access profile:

   [edit access]
   set radius-server 10.20.9.199 port 1812
   set radius-server 10.20.9.199 secret "9Lqa7dsaZjP5F245Fn/0OX7-V24JGDkmf"
   set profile dot1x-auth authentication-order radius
   set profile authp authentication-order radius
   set profile authp radius authentication-server 10.204.96.165
   [edit switch-options]
   set voip interface ge-0/0/8.0 vlan voip

   NOTE: The configured VoIP VLAN cannot be a PVLAN (primary, community, or isolated).

2. Configure the 802.1X settings:

```
[edit interfaces]
set ge-0/0/8 unit 0 family ethernet-switching interface-mode access
set ge-0/0/8 unit 0 family ethernet-switching vlan members data
[edit protocols]
set dot1x authenticator authentication-profile-name authp
set dot1x authenticator interface ge-0/0/8.0 supplicant multiple
set dot1x authenticator interface ge-0/0/8.0 mac-radius
```

NOTE: The configured data VLAN could also be a community VLAN or an isolated VLAN.

3. Configure the VLANs (including the PVLANs):

```
[edit vlans]
set community vlan-id 20
set community private-vlan community
set community-one vlan-id 30
set community-one private-vlan community
set isolated vlan-id 200
set isolated private-vlan isolated
set pvlan vlan-id 2000
set pvlan isolated-vlan isolated
set pvlan community-vlans [community community-one]
set data vlan-id 43
set voip vlan-id 33
```

Results From configuration mode, confirm your configuration by entering the following **show** commands on the switch. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@switch# show access
radius-server {
   10.20.9.199 {
      port 1812;
      secret "$9$Lqa7dsaZjP5F245Fn/0OX7-V24JGDkmf"; ## SECRET-DATA
   }
}
profile dot1x-auth {
   authentication-order radius;
}
profile authp {
   authentication-order radius;
   radius {
      authentication-server 10.204.96.165;
   }
}
user@switch# show interfaces
ge-0/0/8 {
   unit 0 {
      family ethernet-switching {
         interface-mode access;
```

```
        vlan {
          members data;
        }
      }
    }
  }
user@switch# show protocols
dot1x {
  authenticator {
    authentication-profile-name authp;
    interface {
      ge-0/0/8.0 {
        supplicant multiple;
        mac-radius;
      }
    }
  }
}
user@switch# show switch-options
voip {
  interface ge-0/0/8.0 {
    vlan voip;
  }
}
user@switch# show vlans
community {
  vlan-id 20;
  private-vlan community;
}
community-one {
  vlan-id 30;
  private-vlan community;
}
data {
  vlan-id 43;
}
isolated {
  vlan-id 200;
  private-vlan isolated;
}
pvlan {
  vlan-id 2000;
  isolated-vlan isolated;
  community-vlans [community community-one];
}
voip {
  vlan-id 33;
}
```

Verification

- Verify That Client MAC Addresses Are Learned on the Primary VLAN on page 259
- Verify That the Primary VLAN Is an Authenticated VLAN on page 259

Verify That Client MAC Addresses Are Learned on the Primary VLAN

Purpose Show that a client MAC address has been learned on the primary VLAN.

Action

```
user@switch> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static, C - Control MAC, SE - statistics enabled, NM - non configured MAC, R -
remote PE MAC, O - ovsdb MAC)
Ethernet switching table : 1 entries, 1 learned
Routing instance : default-switch
    Vlan        MAC               MAC    Age  Logical      NH     RTR
    name        address           flags       interface    Index  ID
    pvlan       00:30:48:8C:66:BD D      -    ge-0/0/8.0   0      0
```

Verify That the Primary VLAN Is an Authenticated VLAN

Purpose Show that the primary VLAN is shown as an authenticated VLAN.

Action

```
user@switch> show dot1x interface ge-0/0/8.0 detail
ge-0/0/8.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Multiple
  Number of retries: 3
  Quiet period: 60 seconds
  Transmit period: 30 seconds
  Mac Radius: Enabled
  Mac Radius Strict: Disabled
  Reauthentication: Enabled Reauthentication interval: 40 seconds
  Supplicant timeout: 30 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 1
  Guest VLAN member: <not configured>
  Number of connected supplicants: 1
    Supplicant: user5, 00:30:48:8C:66:BD
      Operational state: Authenticated
      Authentication method: Radius
      Authenticated VLAN: pvlan
      Reauthentication due in 17 seconds
```

Putting Access Port Security on Private VLANs

Understanding Access Port Security on PVLANs

You can now enable access port security features, such as DHCP snooping, on private VLANs (PVLANs).

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 PVLAN feature allows you to split a broadcast domain into multiple isolated broadcast subdomains, essentially putting a VLAN inside a VLAN.

Ethernet LANs are vulnerable to attacks such as address spoofing (forging) and Layer 2 denial of service (DoS) on network devices. The following access port security features help protect your device against losses of information and productivity that such attacks can cause, and you can now configure these security features on a PVLAN:

- DHCP snooping—Filters and blocks ingress DHCP server messages on untrusted ports. DHCP snooping builds and maintains a database of DHCP lease information, which is called the DHCP snooping database.
- DHCPv6 snooping—DHCP snooping for IPv6.
- DHCP option 82—Also known as the DHCP Relay Agent Information option. Helps protect the switch against attacks such as spoofing of IP addresses and MAC addresses and DHCP IP address starvation. Option 82 provides information about the network location of a DHCP client. The DHCP server uses this information to implement IP addresses or other parameters for the client.
- DHCPv6 options:
  - Option 37—Remote ID option for DHCPv6; inserts information about the network location of the remote host into DHCPv6 packets.
  - Option 18—Circuit ID option for DHCPv6; inserts information about the client port into DHCPv6 packets.
  - Option 16—Vendor ID option for DHCPv6; inserts information about the vendor of the client hardware into DHCPv6 packets.
- Dynamic ARP inspection (DAI)—Prevents Address Resolution Protocol (ARP) spoofing attacks. ARP requests and replies are compared against entries in the DHCP snooping database, and filtering decisions are made on the basis of the results of those comparisons.
- IP source guard—Mitigates the effects of IP address spoofing attacks on the Ethernet LAN; validates the source IP address in the packet sent from an untrusted access interface against the DHCP snooping database. If the packet cannot be validated, it is discarded.
- IPv6 source guard—IP source guard for IPv6.
- IPv6 neighbor discovery inspection—Prevents IPv6 address spoofing attacks; compares neighbor discovery requests and replies against entries in the DHCPv6 snooping database, and filtering decisions are made on the basis of the results of those comparisons.

NOTE: This document does not provide detailed information about access port security features or PVLANs. For those details, see the feature documentation that is specific to those individual features. For access port security, see Security Services Administration Guide. For PVLANs, see Ethernet Switching Feature Guide.

Configuration Guidelines for Putting Access Port Security Features on PVLANs

Keep the following guidelines and limitations in mind for configuring access port security features on PVLANs:

- You must apply the *same* access port security features on both the primary vlan and all its secondary VLANs.
- A PVLAN can have only one integrated routing and bridging (IRB) interface, and the IRB interface must be on the primary VLAN.
- A trunk port cannot be configured as an untrusted port on an EX4300 switch.
- Limitations on access port security configurations on PVLANs are the same as those for access port security features configurations that are not in PVLANs. See the access port security documentation at Security Services Administration Guide.

Example: Configuring Access Port Security on a PVLAN

Requirements

- Junos OS Release 18.2R1 or later
- EX4300 switch

Overview

The following configuration section shows:

- Configuration of a private VLAN, with the primary VLAN (**vlan-pri**) and its three secondary VLANs—community VLANs (**vlan-hr** and **vlan-finance**) and isolated VLAN (**vlan-iso**).
- Configuration of the interfaces that are used to send communications between the interfaces on those VLANs.

- Configuration of access security features on the primary and secondary VLANs that make up the PVLAN.

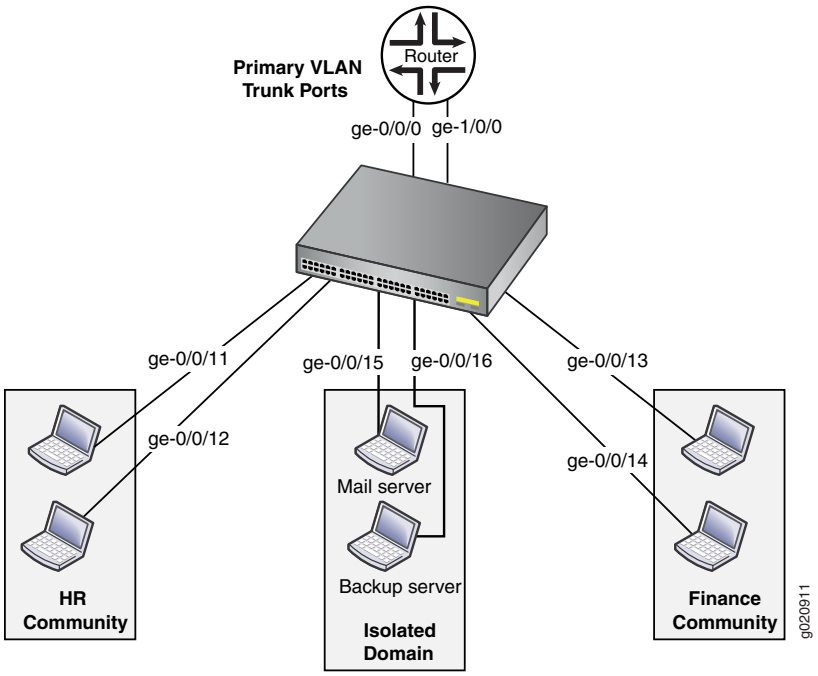


Table 50 on page 262 lists the settings for the example topology.

Table 50: Components of the Topology for Configuring a PVLAN with Access Port Security Features

Interface	Description
ge-0/0/0.0	Primary VLAN (vlan1-pri) trunk interface
ge-0/0/11.0	User 1, HR Community (vlan-hr)
ge-0/0/12.0	User 2, HR Community (vlan-hr)
ge-0/0/13.0	User 3, Finance Community (vlan-finance)
ge-0/0/14.0	User 4, Finance Community (vlan-finance)
ge-0/0/15.0	Mail server, Isolated (vlan-iso)
ge-0/0/16.0	Backup server, Isolated (vlan-iso)
ge-1/0/0.0	Primary VLAN (vlan-pri) trunk interface

Configuring Access Port Security on a PVLAN

CLI Quick Configuration

```
set vlans vlan-pri vlan-id 100
set vlans vlan-hr private-vlan community vlan-id 200
```

```
set vlans vlan-finance private-vlan community vlan-id 300
set vlans vlan-iso private-vlan isolated vlan-id 400
set vlans vlan-pri community-vlan vlan-hr
set vlans vlan-pri community-vlan vlan-finance
set vlans vlan-pri isolated-vlan vlan-iso
set interfaces ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan
  members vlan-hr
set interfaces ge-0/0/12 unit 0 family ethernet-switching interface-mode trunk vlan
  members vlan-hr
set interfaces ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan
  members vlan-finance
set interfaces ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan
  members vlan-finance
set interfaces ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan
  members vlan-iso
set interfaces ge-0/0/16 unit 0 family ethernet-switching interface-mode access vlan
  members vlan-iso
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan
  members vlan-pri
set interfaces ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan
  members vlan-pri
set vlans vlan-pri forwarding-options dhcp-security arp-inspection
set vlans vlan-pri forwarding-options dhcp-security ip-source-guard
set vlans vlan-pri forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-pri forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-pri forwarding-options dhcp-security option-82
set vlans vlan-pri forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-pri forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
set vlans vlan-hr forwarding-options dhcp-security arp-inspection
set vlans vlan-hr forwarding-options dhcp-security ip-source-guard
set vlans vlan-hr forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-hr forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-hr forwarding-options dhcp-security option-82
set vlans vlan-hr forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-hr forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
set vlans vlan-finance forwarding-options dhcp-security arp-inspection
set vlans vlan-finance forwarding-options dhcp-security ip-source-guard
set vlans vlan-finance forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-finance forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-finance forwarding-options dhcp-security option-82
set vlans vlan-finance forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-finance forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
set vlans vlan-iso forwarding-options dhcp-security arp-inspection
set vlans vlan-iso forwarding-options dhcp-security ip-source-guard
set vlans vlan-iso forwarding-options dhcp-security ipv6-source-guard
set vlans vlan-iso forwarding-options dhcp-security neighbor-discovery-inspection
set vlans vlan-iso forwarding-options dhcp-security option-82
set vlans vlan-iso forwarding-options dhcp-security dhcpv6-options option-16
set vlans vlan-iso forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
```

Step-by-Step Procedure

To configure a private VLAN (PVLAN) and then configure access port security features on that PVLAN:

1. Configure the PVLAN—Create the primary VLAN and its secondary VLANs and assign VLAN IDs to them. Associate interfaces with the VLANs. (For details on configuring VLANs, see "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102.)

   1. **[edit vlans]**
      user@switch# set vlan-pri vlan-id 100
      user@switch# set vlan-hr private-vlan community vlan-id 200
      user@switch# set vlan-finance private-vlan community vlan-id 300
      user@switch# set vlan-iso private-vlan isolated vlan-id 400
      user@switch# set vlan-pri community-vlan vlan-hr
      user@switch# set vlan-pri community-vlan vlan-finance
      user@switch# set vlan-pri isolated-vlan vlan-iso

   2. **[edit interfaces]**
      user@switch# set ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan members vlan-hr
      user@switch# set ge-0/0/12 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-hr
      user@switch# set ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan members vlan-finance
      user@switch# set ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-finance
      user@switch# set ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso
      user@switch# set ge-0/0/16 unit 0 family ethernet-switching interface-mode access vlan members vlan-iso
      user@switch# set ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri
      user@switch# set ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members vlan-pri

2. Configure access port security features on the primary VLAN and all its secondary VLANs:

   NOTE: When you configure ARP inspection, IP source guard, IPv6 source guard, neighbor discovery inspection, DHCP option 82, or DHCPv6 options, then DHCP snooping and DHCPv6 snooping are automatically configured.

   [edit vlans]
   user@switch# set vlan-pri forwarding-options dhcp-security arp-inspection
   user@switch# set vlan-pri forwarding-options dhcp-security ip-source-guard
   user@switch# set vlan-pri forwarding-options dhcp-security ipv6-source-guard
   user@switch# set vlan-pri forwarding-options dhcp-security neighbor-discovery-inspection
   user@switch# set vlan-pri forwarding-options dhcp-security option-82
   user@switch# set vlan-pri forwarding-options dhcp-security dhcpv6-options option-16

```
user@switch# set vlan-pri forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
user@switch# set vlan-hr forwarding-options dhcp-security arp-inspection
user@switch# set vlan-hr forwarding-options dhcp-security ip-source-guard
user@switch# set vlan-hr forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-hr forwarding-options dhcp-security
  neighbor-discovery-inspection
user@switch# set vlan-hr forwarding-options dhcp-security option-82
user@switch# set vlan-hr forwarding-options dhcp-security dhcpv6-options
  option-16
user@switch# set vlan-hr forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
user@switch# set vlan-finance forwarding-options dhcp-security arp-inspection
user@switch# set vlan-finance forwarding-options dhcp-security ip-source-guard
user@switch# set vlan-finance forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-finance forwarding-options dhcp-security
  neighbor-discovery-inspection
user@switch# set vlan-finance forwarding-options dhcp-security option-82
user@switch# set vlan-finance forwarding-options dhcp-security dhcpv6-options
  option-16
user@switch# set vlan-finance forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
user@switch# set vlan-iso forwarding-options dhcp-security arp-inspection
user@switch# set vlan-iso forwarding-options dhcp-security ip-source-guard
user@switch# set vlan-iso forwarding-options dhcp-security ipv6-source-guard
user@switch# set vlan-iso forwarding-options dhcp-security
  neighbor-discovery-inspection
user@switch# set vlan-iso forwarding-options dhcp-security option-82
user@switch# set vlan-iso forwarding-options dhcp-security dhcpv6-options
  option-16
user@switch# set vlan-iso forwarding-options dhcp-security dhcpv6-options
  light-weight-dhcpv6-relay
```

Results From configuration mode, confirm your configuration by entering the following **show** commands on the switch. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@switch#  show interfaces
ge-0/0/0 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members vlan-pri;
      }
    }
  }
}
ge-1/0/0 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
```

```
            members vlan-pri;
          }
        }
      }
    }
    ge-0/0/11 {
      unit 0 {
        family ethernet-switching {
          interface-mode access;
          vlan {
            members vlan-hr;
          }
        }
      }
    }
    ge-0/0/12 {
      unit 0 {
        family ethernet-switching {
          interface-mode access;
          vlan {
            members vlan-hr;
          }
        }
      }
    }
    ge-0/0/13 {
      unit 0 {
        family ethernet-switching {
          interface-mode access;
          vlan {
            members vlan-hr;
          }
        }
      }
    }
    ge-0/0/14 {
      unit 0 {
        family ethernet-switching {
          interface-mode access;
          vlan {
            members vlan-hr;
          }
        }
      }
    }
    ge-0/0/15 {
      unit 0 {
        family ethernet-switching {
          interface-mode access;
          vlan {
            members vlan-iso;
          }
        }
      }
    }
    ge-0/0/16 {
```

```
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan {
          members vlan-iso;
        }
      }
    }
}
user@switch#  show vlans
vlan-finance {
    vlan-id 300;
    private-vlan community;
    interface {
      ge-0/0/13.0;
      ge-0/0/14.0;
    }
    forwarding-options {
      dhcp-security {
        arp-inspection;
          ip-source-guard;
          neighbor-discovery-inspection;
          ipv6-source-guard;
          option-82;
          dhcpv6-options light-weight-dhcpv6-relay;
        dhcpv6-options option-16;
      }
    }
}
vlan-hr {
    vlan-id 200;
    private-vlan community;
    interface {
      ge-0/0/11.0;
      ge-0/0/12.0;
    }
    forwarding-options {
      dhcp-security {
        arp-inspection;
          ip-source-guard;
          neighbor-discovery-inspection;
          ipv6-source-guard;
          option-82;
          dhcpv6-options light-weight-dhcpv6-relay;
        dhcpv6-options option-16;
      }
    }
}
vlan-iso {
    vlan-id 400;
    private-vlan isolated;
    interface {
      ge-0/0/15.0;
      ge-0/0/16.0;
    }
    forwarding-options {
```

```
    dhcp-security {
      arp-inspection;
        ip-source-guard;
        neighbor-discovery-inspection;
        ipv6-source-guard;
        option-82;
        dhcpv6-options light-weight-dhcpv6-relay;
      dhcpv6-options option-16;
    }
  }
}
vlan-pri {
  vlan-id 100;
    community-vlan vlan-finance;
    community-vlan vlan-hr;
  isolated-vlan vlan-iso;
  interface {
    ge-0/0/0.0;
    ge-1/0/0.0;
  }
  forwarding-options {
    dhcp-security {
      arp-inspection;
        ip-source-guard;
        neighbor-discovery-inspection;
        ipv6-source-guard;
        option-82;
        dhcpv6-options light-weight-dhcpv6-relay;
      dhcpv6-options option-16;
    }
  }
}
```

Verification

Verify That Access Security Features Are Working as Expected

Purpose Verify that the access port security features that you configured on your PVLAN are working as expected.

Action Use the **show dhcp-security** and the **clear dhcp-security** CLI commands to verify that the features are working as expected. See details about those commands in Security Services Administration Guide.

Creating a Private VLAN on a Single QFX 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 226
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- Verifying That a Private VLAN Is Working on a Switch on page 355

Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)

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 on EX Series switches enables you to split a broadcast domain, also known as a primary VLAN, into multiple isolated broadcast subdomains, also known as secondary VLANs. Splitting the primary VLAN into secondary VLANs essentially nests a VLAN inside another 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. (Unlike the secondary VLANs, you do not need to preconfigure the primary VLAN—this procedure provides the complete configuration of the primary VLAN.) Although tags are not needed when a secondary VLAN is configured on a single switch, configuring a secondary VLAN as tagged does not adversely affect its functionality. For instructions on configuring the secondary VLANs, see "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98.

Keep these rules in mind when configuring a PVLAN on a single switch:

- The primary VLAN must be a tagged VLAN.
- Configuring a VoIP VLAN on PVLAN interfaces is not supported.

To configure a private VLAN on a single switch:

1. Set the VLAN ID for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id vlan-id-number
   ```

2. Set the interfaces and port modes:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching port-mode mode
   user@switch# set interface-name unit 0 family ethernet-switching vlan members (all | vlan-id
   | vlan-number)
   ```

3. Configure the access ports in the primary VLAN to not forward packets to one another:

   ```
   [edit vlans]
   user@switch# set vlan-id vlan-id-number no-local-switching
   ```

4. For each community VLAN, configure access interfaces:

   ```
   [edit vlans]
   user@switch# set community-vlan-name interface-mac-limit interface-name
   ```

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

   ```
   [edit vlans]
   user@switch# set community-vlan-name primary-vlan primary-vlan-name
   ```

Isolated VLANs are not configured as part of this process. Instead, they are created internally if **no-local-switching** is enabled on the primary VLAN and the isolated VLAN has access interfaces as members.

To optionally enable routing between isolated and community VLANs by using a routed VLAN interface (RVI) instead of a promiscuous port connected to a router, see "Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch (CLI Procedure)" on page 341.

NOTE: Only an EX8200 switch or EX8200 Virtual Chassis support the use of an RVI to route Layer 3 traffic between isolated and community VLANs in a PVLAN domain.

Related Documentation

- Example: Configuring a Private VLAN on a Single EX Series Switch on page 284
- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*
- Verifying That a Private VLAN Is Working on a Switch on page 355
- Understanding Private VLANs on page 226

Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure)

NOTE: This task uses Junos OS for switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your EX Series switch runs software that does not support ELS, see "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 271. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

NOTE: Private VLANs are not supported on QFX5100 switches and QFX10002 switches running Junos OS Release 15.1X53.

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic or limit the communication between known hosts. Private VLANs (PVLANs) enable you to split a broadcast domain (primary VLAN) into multiple isolated broadcast subdomains (secondary VLANs), essentially putting a VLAN inside a VLAN. This procedure describes how to create a PVLAN on a single switch.

NOTE: You must specify a VLAN ID for each secondary VLAN even if the PVLAN is configured on a single switch.

You do not need to preconfigure the primary VLAN. This topic shows the primary VLAN being configured as part of this PVLAN configuration procedure.

For a list of guidelines on configuring PVLANs, see "Understanding Private VLANs" on page 226.

To configure a private VLAN on a single switch:

1. Set the VLAN ID for the primary VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id vlan-id-number
   ```

2. Configure at least one interface within the primary VLAN so that it communicates with all the subdomains of the PVLAN. This interface functions as a *promiscuous* port. It can be either a trunk port or an access port.

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching
   user@switch# set interface-name unit 0 family ethernet-switching vlan members
   primary-vlan-name
   ```

3. Configure another promiscuous interface of the primary VLAN as a trunk port to connect the PVLAN to the external router or switch:

   ```
   [edit interfaces]
   ```

```
user@switch#  set interface-name unit 0 family ethernet-switching interface-mode trunk
user@switch#  set interface-name unit 0 family ethernet-switching vlan members
primary-vlan-name
```

4. Create an isolated VLAN by selecting the **isolated** option for **private-vlan**, and setting a VLAN ID for the isolated VLAN:

```
[edit vlans]
user@switch#  set isolated-vlan-name private-vlan isolated vlan-id isolated-vlan-id
```

NOTE: You can create only one isolated VLAN within a private VLAN. Setting the VLAN name for the isolated VLAN is optional. Configuring the VLAN ID is required.

5. Create a community VLAN by selecting the **community** option for **private-vlan**, and setting a VLAN ID for this community VLAN:

```
[edit vlans]
user@switch#  set community-vlan-name private-vlan community vlan-id community-vlan-id
```

NOTE: To create additional community VLANs, repeat this step and specify a different name for the community VLAN. Setting the VLAN name for the community VLAN is optional. Configuring the VLAN ID is required.

6. Associate the isolated VLAN with the primary VLAN:

```
[edit vlans]
user@switch#  set primary-vlan-name vlan-id primary-vlan-id isolated-vlan isolated-vlan-name
```

7. Associate each community VLAN with the primary VLAN:

```
[edit vlans]
user@switch#  set primary-vlan-name vlan-id primary-vlan-id
community-vlan community-vlan-name
```

8. If you have not already done so, configure at least one interface of the isolated VLAN.

```
[edit interfaces]
user@switch#  set interface-name unit logical-unit-number family ethernet-switching
interface-mode access vlan members isolated-vlan-name
```

9. If you have not already done so, configure at least one interface of the community VLAN.

```
[edit interfaces]
user@switch#  set interface-name unit logical-unit-number family ethernet-switching
interface-mode access vlan members community-vlan-name
```

NOTE: Repeat the same step on other community VLANs that you want to include in the PVLAN.

Related Documentation

- Understanding Private VLANs on page 226
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277

Creating a Private VLAN Spanning Multiple QFX Series 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 226
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241
- Creating a Private VLAN on a Single QFX Switch on page 269
- Verifying That a Private VLAN Is Working on a Switch on page 355

Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style If your switch runs software that does not support ELS, see *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

NOTE: Private VLANs are not supported on QFX5100 switches and QFX10002 switches running Junos OS Release 15.1X53.

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic or limit the communication between known hosts. Private VLANs (PVLANs) enable you to split a broadcast domain (primary VLAN) into multiple isolated broadcast subdomains (secondary VLANs), essentially putting a VLAN inside a VLAN This procedure describes how to configure a PVLAN to span multiple switches.

For a list of guidelines on configuring PVLANs, see "Understanding Private VLANs" on page 226.

To configure a PVLAN to span multiple switches, perform the following procedure on all the switches that will participate in the PVLAN::

1. Create the primary VLAN by setting the unique VLAN name and specify an 802.1Q tag for the VLAN:

   ```
   [edit vlans]
   user@switch# set primary-vlan-name vlan-id number
   ```

2. On the switch that will connect to a router, configure a promiscuous interface as a trunk port to connect the PVLAN to the router:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching interface-mode trunk
   user@switch# set interface-name unit 0 family ethernet-switching vlan members
   primary-vlan-name
   ```

3. On all the switches, configure a trunk interface as the Inter-Switch Link (ISL) that will be used to connect the switches to each other:

   ```
   [edit interfaces]
   user@switch# set interface-name unit 0 family ethernet-switching interface-mode trunk
   inter-switch-link
   user@switch# set interface-name unit 0 family ethernet-switching vlan members
   name-of-private-vlan
   ```

4. Create an isolated VLAN within the primary VLAN by selecting the **isolated** option for **private-vlan**, and setting a VLAN ID for the isolated VLAN:

```
[edit vlans]
user@switch# set isolated-vlan-name private-vlan isolated vlan-id isolated-vlan-id
```

NOTE: You can create only one isolated VLAN within a private VLAN. The isolated VLAN can contain member interfaces from the multiple switches that compose the PVLAN.
Setting the VLAN name for the isolated VLAN is optional. Configuring the VLAN ID is required.

5. Create a community VLAN within the primary VLAN by selecting the **community** option for **private-vlan**, and setting a VLAN ID for this community VLAN::

```
[edit vlans]
user@switch# set community-vlan-name private-vlan community vlan-id community-vlan-id
```

NOTE: To create additional community VLANs, repeat this step and specify a different name for the community VLAN. Setting the VLAN name for the community VLAN is optional. Configuring the VLAN ID is required.

6. Associate the isolated VLAN with the primary VLAN:

```
[edit vlans primary-vlan-name vlan-id primary-vlan-id]
user@switch# set isolated-vlanisolated-vlan-name
```

7. Associate each community VLAN with the primary VLAN:

```
[edit vlans primary-vlan-name vlan-id primary-vlan-id]
user@switch# set community-vlancommunity-vlan-name
```

8. If you have not already done so, configure at least one access interface to be a member of the isolated VLAN.

```
[edit interface]
user@switch# set interface-name unit logical-unit-number family ethernet-switching
interface-mode access vlan members isolated-vlan-name
```

9. If you have not already done so, configure at least one access interface to be a member of the community VLAN.

```
[edit interface]
user@switch# set interface-name unit logical-unit-number family ethernet-switching
interface-mode access vlan members community-vlan-name
```

NOTE: Repeat this step for the other community VLANs that you are including in the PVLAN.

Related Documentation

- Understanding Private VLANs on page 226
- Example: Configuring a Private VLAN on a Single Switch with ELS Support on page 291
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 273

Example: Configuring a Private VLAN on a Single QFX Series 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 Switches" on page 93.

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 51 on page 279 lists the settings for the sample topology.

Table 51: 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**)

Table 51: Components of the Topology for Configuring a PVLAN (continued)

Interface	Description
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 283

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 226
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241
- Creating a Private VLAN on a Single QFX Switch on page 269

Example: Configuring a Private VLAN on a Single EX Series 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 on EX Series switches 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 EX Series switch:

NOTE: Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 9.3 or later for EX Series switches

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98.

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.

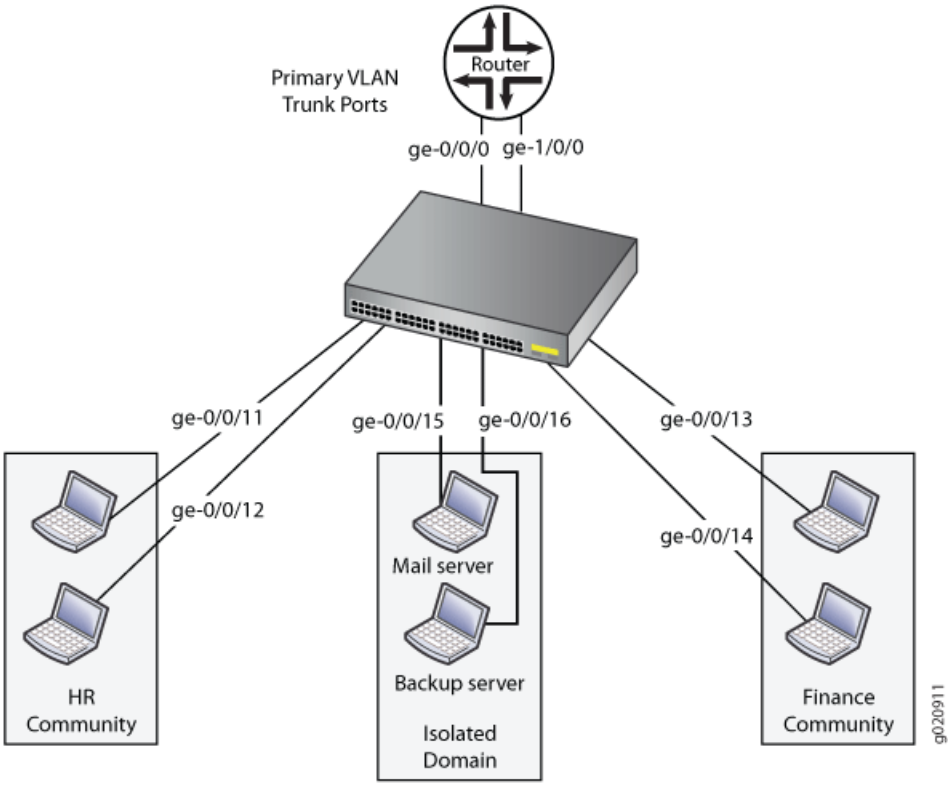
Table 51 on page 279 lists the settings for the example topology.

Table 52: Components of the Topology for Configuring a PVLAN

Interface	Description
ge-0/0/0.0	Primary VLAN (**vlan1**) 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 (**pvlan**) trunk interface

Figure 17 on page 286 shows the topology for this example.

Figure 17: Topology of a Private VLAN on a Single EX Series Switch



Configuration

To configure a PVLAN, perform these tasks:

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 vlan1 vlan-id 1000
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 vlan1
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 vlan1
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 vlan1 no-local-switching
set vlans vlan1 interface ge-0/0/0.0
set vlans vlan1 interface ge-1/0/0.0
set vlans hr-comm vlan-id 400
set vlans hr-comm interface ge-0/0/11.0
set vlans hr-comm interface ge-0/0/12.0
```

```
set vlans finance-comm vlan-id 300
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 vlan1
set vlans finance-comm primary-vlan vlan1
set vlans vlan1 interface ge-0/0/15.0
set vlans vlan1 interface ge-0/0/16.0
```

Step-by-Step Procedure

To configure the PVLAN:

1. Set the VLAN ID for the primary VLAN:

```
[edit vlans]
user@switch# set vlan1 vlan-id 1000
```

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 vlan1
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 vlan1 no-local-switching
```

4. Add the trunk interfaces to the primary VLAN:

```
[edit vlans]
user@switch# set vlan1 interface ge-0/0/0.0
user@switch# set vlan1 interface ge-1/0/0.0
```

5. For each secondary VLAN, configure the VLAN IDs and the 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 vlan-id 400
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 vlan-id 300
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 vlan1
user@switch#  set finance-comm primary-vlan vlan1
```

7. Add each isolated interface to the primary VLAN:

```
[edit vlans]
user@switch#  set vlan1 interface ge-0/0/15.0
user@switch#  set vlan1 interface ge-0/0/16.0
```

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 vlan1;
            }
         }
      }
   }
   ge-1/0/0 {
      unit 0 {
         family ethernet-switching {
            port-mode trunk;
            vlan {
               members vlan1;
            }
         }
      }
   }
   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 {
        vlan-id 300;
        interface {
          ge-0/0/13.0;
          ge-0/0/14.0;
        }
        primary-vlan vlan1;
      }
      hr-comm {
        vlan-id 400;
        interface {
          ge-0/0/11.0;
          ge-0/0/12.0;
        }
        primary-vlan vlan1;
      }
      vlan1 {
        vlan-id 1000;
        interface {
          ge-0/0/15.0;
          ge-0/0/16.0;
          ge-0/0/0.0;
          ge-1/0/0.0;
        }
        no-local-switching;
      }
    }
```

Verification

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

- Verifying That the Private VLAN and Secondary VLANs Were Created on page 290

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 vlan1 extensive
VLAN: vlan1, Created at: Tue Sep 16 17:59:47 2008
802.1Q Tag: 1000, 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 :
      __vlan1_vlan1_ge-0/0/15.0__
      __vlan1_vlan1_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
802.1Q Tag: 400,Internal index: 22, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: vlan1
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
802.1Q Tag: 300,Internal index: 21, Admin State: Enabled, Origin: Static
Private VLAN Mode: Community, Primary VLAN: vlan1
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 __vlan1_vlan1_ge-0/0/15.0__ extensive
VLAN: __vlan1_vlan1_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: vlan1
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 __vlan1_vlan1_ge-0/0/16.0__ extensive
VLAN: __vlan1_vlan1_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: vlan1
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

- Example: Configuring a Private VLAN Spanning Multiple EX Series Switches on page 326
- Creating a Private VLAN on a Single EX Series Switch (CLI Procedure) on page 271

Example: Configuring a Private VLAN on a Single Switch with ELS Support

NOTE: This example uses Junos OS for switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your EX switch runs software that does not support ELS, see "Example: Configuring a Private VLAN on a Single EX Series Switch" on page 284. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

NOTE: Private VLANs are not supported on QFX5100 switches and QFX10002 switches running Junos OS Release 15.1X53.

For security reasons, it is often useful to restrict the flow of broadcast and unknown unicast traffic or limit the communication between known hosts. Private VLANs (PVLANs) enable you to split a broadcast domain (primary VLAN) into multiple isolated broadcast subdomains (secondary VLANs), 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 Junos OS switch

- Junos OS Release 14.1X53-D10 or later for EX Series switches

  Junos OS Release 14.1X53-D15 or later for QFX Series switches

Overview and Topology

You can isolate groups of subscribers for improved security and efficiency. This configuration example uses a simple topology to illustrate how to create a PVLAN with one primary VLAN and three secondary VLANs (one isolated VLAN, and two community VLANs).

Table 51 on page 279 lists the interfaces of the topology used in the example.

Table 53: Interfaces of the Topology for Configuring a PVLAN

Interface	Description
ge-0/0/0 ge-1/0/0	Promiscuous member ports
ge-0/0/11, ge-0/0/12	HR community VLAN member ports
ge-0/0/13, ge-0/0/14	Finance community VLAN member ports
ge-0/0/15, ge-0/0/16	Isolated member ports

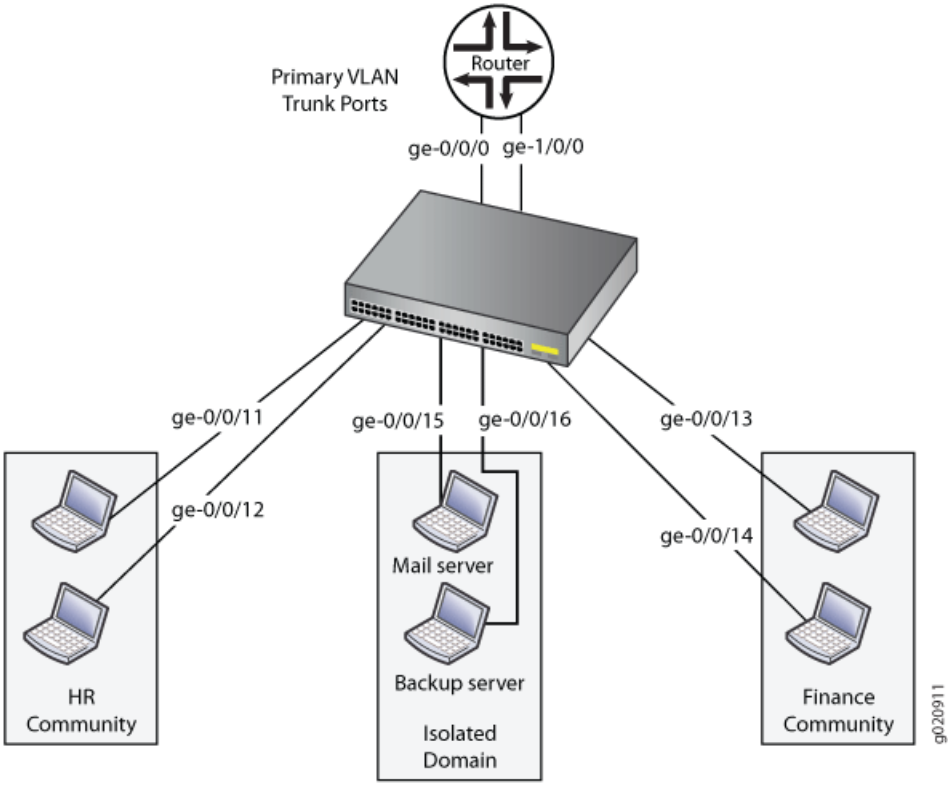
Table 54 on page 292 lists the VLAN IDs of the topology used in the example.

Table 54: VLAN IDs in the Topology for Configuring a PVLAN

VLAN ID	Description
100	Primary VLAN
200	HR community VLAN
300	Finance community VLAN
400	Isolated VLAN

Figure 18 on page 293 shows the topology for this example.

Figure 18: Topology of a Private VLAN on a Single EX Series Switch



Configuration

You can use an existing VLAN as the basis for your private PVLAN and create subdomains within it. This example creates a primary VLAN—using the VLAN name **vlan-pri**—as part of the procedure.

To configure a PVLAN, perform these tasks:

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 vlan-pri vlan-id 100
set vlans vlan-iso private-vlan isolated vlan-id 400
set vlans vlan-hr private-vlan community vlan-id 200
set vlans vlan-finance private-vlan community vlan-id 300
set vlans vlan-pri vlan-id 100 isolated-vlan vlan-iso community-vlan vlan-hr community-vlan
vlan-finance
set interface ge-0/0/11 unit 0 family ethernet-switching interface-mode access vlan members
vlan-hr
set interface ge-0/0/12 unit 0 family ethernet-switching interface-mode trunk vlan members
vlan-hr
set interface ge-0/0/13 unit 0 family ethernet-switching interface-mode access vlan members
vlan-finance
set interface ge-0/0/14 unit 0 family ethernet-switching interface-mode trunk vlan members
vlan-finance
```

```
set interface ge-0/0/15 unit 0 family ethernet-switching interface-mode access vlan members
vlan-iso
set interface ge-0/0/16 unit 0 family ethernet-switching interface-mode access vlan members
vlan-iso
set interface ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members
vlan-pri
set interface ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan members
vlan-pri
```

Step-by-Step Procedure

To configure the PVLAN:

1. Create the primary VLAN (in this example, the name is **vlan-pri**) of the private VLAN:

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

2. Create an isolated VLAN and assign it a VLAN ID:

   ```
   [edit vlans]
   user@switch# set vlan-iso private-vlan isolated vlan-id 400
   ```

3. Create the HR community VLAN and assign it a VLAN ID:

   ```
   [edit vlans]
   user@switch# set vlan-hr private-vlan community vlan-id 200
   ```

4. Create the finance community VLAN and assign it a VLAN ID:

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

5. Associate the secondary VLANs with the primary VLAN:

   ```
   [edit vlans]
   user@switch# set vlan-pri vlan-id 100 isolated-vlan vlan-iso community-vlan vlan-hr
   community-vlan vlan-finance
   ```

6. Set the interfaces to the appropriate interface modes:

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

7. Configure a promiscuous trunk interface of the primary VLAN. This interface is used by the primary VLAN to communicate with the secondary VLANs.

```
user@switch# set ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan
members vlan-pri
```

8. Configure another trunk interface (it is also a promiscuous interface) of the primary VLAN, connecting the PVLAN to the router.

```
user@switch# set ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk vlan
members vlan-pri
```

Results

Check the results of the configuration:

```
user@switch> show configuration
```

Verification

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

- Verifying That the Private VLAN and Secondary VLANs Were Created on page 295

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 extensive
```

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 226
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277

Example: Configuring a Private VLAN Spanning Multiple QFX 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 Switches" on page 93.

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 226.

Figure 19 on page 297 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 19: PVLAN Topology Spanning Multiple Switches

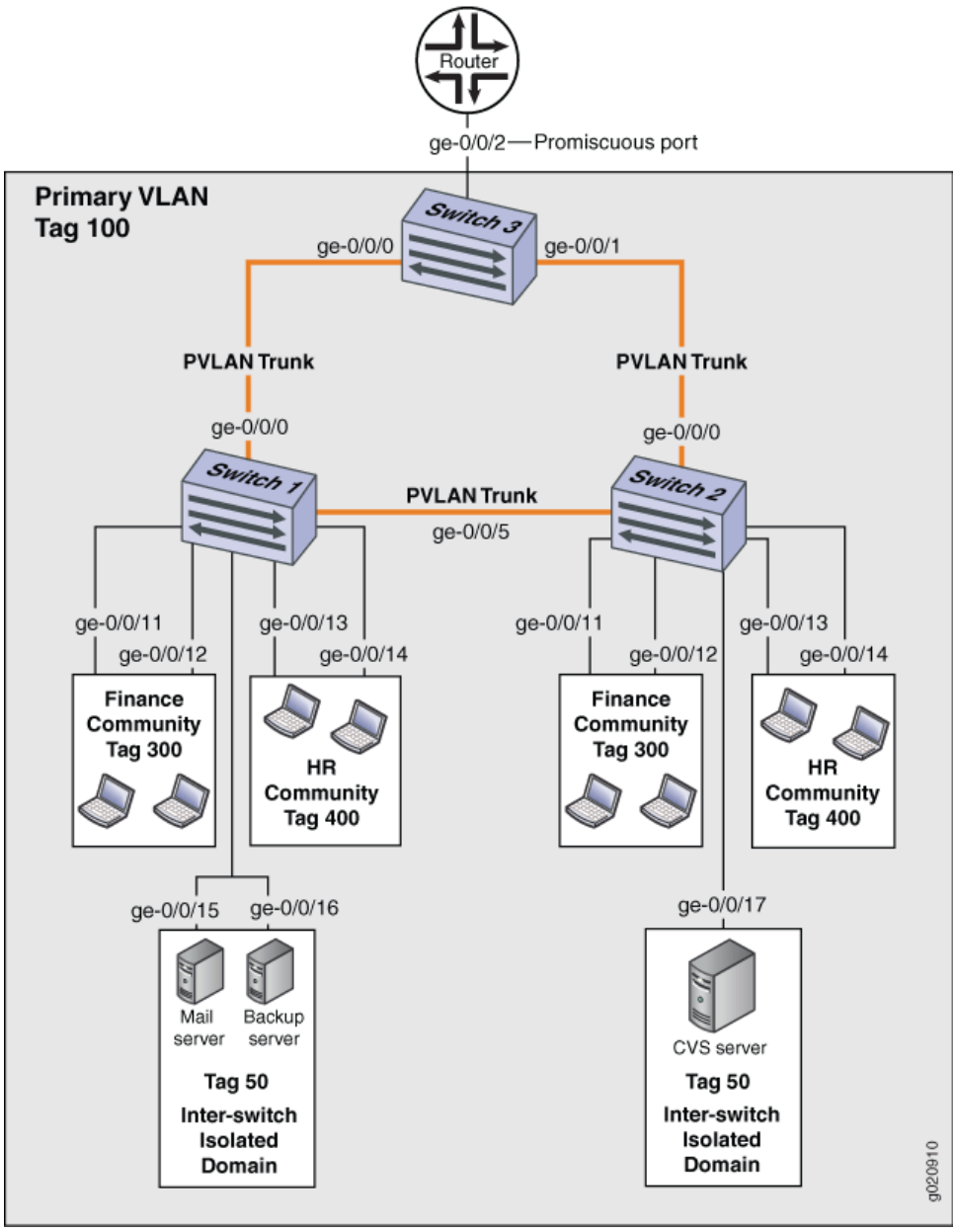


Table 55 on page 297, Table 56 on page 298, and Table 57 on page 298 list the settings for the example topology.

Table 55: 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**

Table 55: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple Devices (continued)

Property	Settings
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 56: 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 57: 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

Table 57: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple Devices (continued)

Property	Settings
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 306
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2 on page 308
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3 on page 309

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 226
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241
- Example: Configuring a Private VLAN on a Single QFX Series Switch on page 279

Example: Configuring a Private VLAN Spanning Multiple Switches With an IRB Interface

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.

Just like regular VLANs, PVLANs are isolated at Layer 2 and normally require that a Layer 3 device be used if you want to route traffic. Starting with Junos OS 14.1X53-D30, you can use an integrated routing and bridging (IRB) interface to route Layer 3 traffic between devices connected to a PVLAN. Using an IRB interface in this way can also allow the devices in the PVLAN to communicate at Layer 3 with devices in other community or

isolated VLANs or with devices outside the PVLAN. This example also demonstrates how to include an IRB interface in a PVLAN configuration.

Requirements

This example uses the following hardware and software components:

- Three QFX Series or EX4600 switches
- Junos OS release with PVLAN for QFX Series or EX4600

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 switches, 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 devices in the PVLAN are connected at Layer 3 to each other and to devices outside the PVLAN through an IRB interface 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 226.

Figure 20 on page 312 shows the topology for this example.

Figure 20: PVLAN Topology Spanning Multiple Switches with an IRB Interface

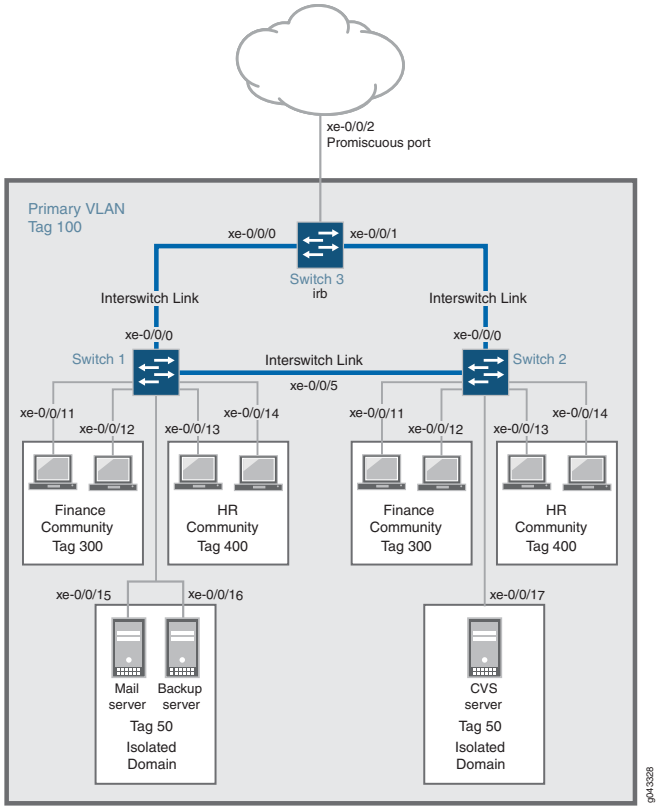


Table 55 on page 297, Table 56 on page 298, and Table 57 on page 298 list the settings for the example topology.

Table 58: 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** **isolated-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
Interswitch link interfaces	**xe-0/0/0.0**, connects Switch 1 to Switch 3 **xe-0/0/5.0**, connects Switch 1 to Switch 2
Isolated Interfaces in primary VLAN	**xe-0/0/15.0**, mail server **xe-0/0/16.0**, backup server
Interfaces in VLAN **finance-com**	**xe-0/0/11.0** **xe-0/0/12.0**

Table 58: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple Devices (continued)

Property	Settings
Interfaces in VLAN **hr-comm**	**xe-0/0/13.0** **xe-0/0/14.0**

Table 59: 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** **isolated-vlan-id**, tag **50** **finance-comm**, tag **300** **hr-comm**, tag **400**
Interswitch link interfaces	**xe-0/0/0.0**, connects Switch 2 to Switch 3 **xe-0/0/5.0**, connects Switch 2 to Switch 1
Isolated Interface in primary VLAN	**xe-0/0/17.0**, CVS server
Interfaces in VLAN **finance-com**	**xe-0/0/11.0** **xe-0/0/12.0**
Interfaces in VLAN **hr-comm**	**xe-0/0/13.0** **xe-0/0/14.0**

Table 60: 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** **isolated-vlan-id**, tag **50finance-comm**, tag **300** **hr-comm**, tag **400**
Interswitch link interfaces	**xe-0/0/0.0**, connects Switch 3 to Switch 1. **xe-0/0/1.0**, connects Switch 3 to Switch 2.
Promiscuous port	**xe-0/0/2**, connects the PVLAN to another network. 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.

Table 60: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple Devices (continued)

Property	Settings
IRB interface	**xe-0/0/0** **xe-0/0/1** Configure unrestricted proxy ARP on the IRB interface to allow ARP resolution to occur so that devices that use IPv4 can communicate at Layer 3. For IPv6 traffic, you must explicitly map an IRB address to the destination address to allow ARP resolution.

Configuration Overview

When configuring a PVLAN on multiple switches, the following rules apply:

- The primary VLAN must be a tagged VLAN.
- The primary VLAN is the only VLAN that can be a member of an interswitch link interface.

When configuring an IRB interface in a PVLAN, these rules apply:

- You can create only one IRB interface in a PVLAN, regardless of how many switches participate in the PVLAN.
- The IRB interface must be a member of the primary VLAN in the PVLAN.
- Each host device that you want to connect at Layer 3 must use an IP address of the IRB as its default gateway address.

Configuring a PVLAN on Switch 1

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 interfaces xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members 100
set interfaces xe-0/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/5 unit 0 family ethernet-switching vlan members 100
set vlans finance-comm vlan-id 300 private-vlan community
set vlans hr-comm vlan-id 400 private-vlan community
set vlans isolated-vlan vlan-id 50 private-vlan isolated
set vlans pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members 400
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members 400
set interfaces xe-0/0/15 unit 0 family ethernet-switching vlan members 50
set interfaces xe-0/0/16 unit 0 family ethernet-switching vlan members 50
```

Step-by-Step Procedure

1. Configure interface xe-0/0/0 to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

2. Configure interface xe-0/0/0 to be an interswitch link that carries all the VLANs:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
   ```

3. Configure pvlan100 (the primary VLAN) to be a member of interface xe-0/0/0:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching vlan members 100
   ```

4. Configure interface xe-0/0/5 to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

5. Configure interface xe-0/0/5 to be an interswitch link that carries all the VLANs:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
   ```

6. Configure pvlan100 to be a member of interface xe-0/0/5:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching vlan members 100
   ```

7. Create the community VLAN for the finance organization:

   ```
   [edit vlans]
   set finance-comm vlan-id 300 private-vlan community
   ```

8. Create the community VLAN for the HR organization:

   ```
   [edit vlans]
   set hr-comm vlan-id 400 private-vlan community
   ```

9. Create the isolated VLAN for the mail and backup servers:

   ```
   [edit vlans]
   set isolated-vlan vlan-id 50 private-vlan isolated
   ```

10. Create the primary VLAN and make the community and isolated VLANs members of it:

    ```
    [edit vlans]
    set pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50
    ```

11. Configure VLAN 300 (the a community VLAN) to be a member of interface xe-0/0/11:

    ```
    [edit interfaces]
    user@switch# set xe-0/0/11 unit 0 family ethernet-switching vlan members 300
    ```

12. Configure VLAN 300 (a community VLAN) to be a member of interface xe-0/0/12:

```
[edit interfaces]
user@switch# set xe-0/0/12 unit 0 family ethernet-switching vlan members 300
```

13. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/13:

```
[edit interfaces]
user@switch# set xe-0/0/13 unit 0 family ethernet-switching vlan members 400
```

14. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/14:

```
[edit interfaces]
user@switch# set xe-0/0/14 unit 0 family ethernet-switching vlan members 400
```

15. Configure VLAN 50 (the isolated VLAN) to be a member of interface xe-0/0/15:

```
[edit interfaces]
user@switch# set xe-0/0/15 unit 0 family ethernet-switching vlan members 50
```

16. Configure VLAN 50 (the isolated VLAN) to be a member of interface xe-0/0/16:

```
[edit interfaces]
user@switch# set xe-0/0/16 unit 0 family ethernet-switching vlan members 50
```

Results

Check the results of the configuration:

```
[edit]
  user@switch# show
vlans {
  finance-comm {
    vlan-id 300;
    private-vlan community;
  }
  hr-comm {
    vlan-id 400;
    private-vlan community;
  }
  isolated-vlan{
    vlan-id 50;
    private-vlan isolated;
  }
  pvlan100 {
    vlan-id 100;
    isolated-vlan 50;
    community-vlans [300 400]
  }
}
```

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 isolated VLAN. For Switch 2, the isolated VLAN interface is xe-0/0/17.0 .

```
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members 100
set interfaces xe-0/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/5 unit 0 family ethernet-switching vlan members 100
set vlans finance-comm vlan-id 300 private-vlan community
set vlans hr-comm vlan-id 400 private-vlan community
set vlans isolated-vlan vlan-id 50 private-vlan isolated
set vlans pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/12 unit 0 family ethernet-switching vlan members 300
set interfaces xe-0/0/13 unit 0 family ethernet-switching vlan members 400
set interfaces xe-0/0/14 unit 0 family ethernet-switching vlan members 400
set interfaces xe-0/0/17 unit 0 family ethernet-switching vlan members 50
```

Step-by-Step Procedure

1. Configure interface xe-0/0/0 to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

2. Configure interface xe-0/0/0 to be an interswitch link that carries all the VLANs:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
   ```

3. Configure pvlan100 (the primary VLAN) to be a member of interface xe-0/0/0:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching vlan members 100
   ```

4. Configure interface xe-0/0/5 to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

5. Configure interface xe-0/0/5 to be an interswitch link that carries all the VLANs:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
   ```

6. Configure pvlan100 to be a member of interface xe-0/0/5:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching vlan members 100
   ```

7. Create the community VLAN for the finance organization:

   ```
   [edit vlans]
   set finance-comm vlan-id 300 private-vlan community
   ```

8. Create the community VLAN for the HR organization:

   ```
   [edit vlans]
   set hr-comm vlan-id 400 private-vlan community
   ```

9. Create the isolated VLAN for the mail and backup servers:

   ```
   [edit vlans]
   set isolated-vlan vlan-id 50 private-vlan isolated
   ```

10. Create the primary VLAN and make the community and isolated VLANs members of it:

    ```
    [edit vlans]
    set pvlan100 vlan-id 100 community-vlans [300 400] isolated-vlan 50
    ```

11. Configure VLAN 300 (the a community VLAN) to be a member of interface xe-0/0/11:

    ```
    [edit interfaces]
    user@switch# set xe-0/0/11 unit 0 family ethernet-switching vlan members 300
    ```

12. Configure VLAN 300 (a community VLAN) to be a member of interface xe-0/0/12:

    ```
    [edit interfaces]
    user@switch# set xe-0/0/12 unit 0 family ethernet-switching vlan members 300
    ```

13. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/13:

    ```
    [edit interfaces]
    user@switch# set xe-0/0/13 unit 0 family ethernet-switching vlan members 400
    ```

14. Configure VLAN 400 (a community VLAN) to be a member of interface xe-0/0/14:

    ```
    [edit interfaces]
    user@switch# set xe-0/0/14 unit 0 family ethernet-switching vlan members 400
    ```

15. Configure VLAN 50 (the isolated VLAN) to be a member of interface xe-0/0/17:

    ```
    [edit interfaces]
    user@switch# set xe-0/0/17 unit 0 family ethernet-switching vlan members 50
    ```

Results

Check the results of the configuration:

```
[edit]
  user@switch# show
vlans {
  finance-comm {
    vlan-id 300;
    private-vlan community;
  }
  hr-comm {
    vlan-id 400;
    private-vlan community;
  }
  isolated-vlan{
    vlan-id 50;
    private-vlan isolated;
  }
  pvlan100 {
    vlan-id 100;
    isolated-vlan 50;
    community-vlans [300 400]
  }
}
```

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 xe-0/0/2.0 is a trunk port connecting the PVLAN to another network.

```
[edit]
[edit]
set interfaces xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/0 unit 0 family ethernet-switching vlan members 100
set interfaces xe-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/1 unit 0 family ethernet-switching inter-switch-link
set interfaces xe-0/0/1 unit 0 family ethernet-switching vlan members 100
set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/2 unit 0 family ethernet-switching vlan members 100
set vlans pvlan100 vlan-id 100
set interfaces irb unit 100 family inet address 192.168.1.1/24
set vlans pvlan100 l3-interface irb.100
set interfaces irb unit 100 proxy-arp unrestricted
```

Step-by-Step Procedure

To configure Switch 3 to function as the distribution switch for this PVLAN, use the following procedure:

1. Configure interface xe-0/0/0 to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

2. Configure interface xe-0/0/0 to be an interswitch link that carries all the VLANs:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching inter-switch-link
   ```

3. Configure pvlan100 (the primary VLAN) to be a member of interface xe-0/0/0:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching vlan members 100
   ```

4. Configure interface xe-0/0/5 to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/0 unit 0 family ethernet-switching interface-mode trunk
   ```

5. Configure interface xe-0/0/5 to be an interswitch link that carries all the VLANs:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching inter-switch-link
   ```

6. Configure pvlan100 to be a member of interface xe-0/0/5:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/5 unit 0 family ethernet-switching vlan members 100
   ```

7. Configure interface xe-0/0/2 (the promiscuous interface) to be a trunk:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk
   ```

8. Configure pvlan100 to be a member of interface xe-0/0/2:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/2 unit 0 family ethernet-switching vlan members 100
   ```

9. Create the primary VLAN:

   ```
   [edit vlans]
   set vlans pvlan100 vlan-id 100
   ```

10. Create the IRB interface **irb** and assign it an address in the subnet used by the devices attached to Switches 1 and 2:

    ```
    [edit interfaces]
    set irb unit 100 family inet address 192.168.1.1/24
    ```

NOTE: Each host device that you want to connect at Layer 3 must be in the same subnet as the IRB interface and use the IP address of the IRB interface as its default gateway address.

11. Complete the IRB interface configuration by binding the interface to the primary VLAN **pvlan100**:

```
[edit vlans]
set pvlan100 l3-interface irb.100
```

12. Configure unrestricted proxy ARP for each unit of the IRB interface so that ARP resolution works for IPv4 traffic:

```
[edit interfaces]
set irb unit 100 proxy-arp unrestricted
```

NOTE: Because the devices in the community and isolated VLANs are isolated at Layer 2, this step is required to allow ARP resolution to occur between the VLANs so that devices using IPv4 can communicate at Layer 3. (For IPv6 traffic, you must explicitly map an IRB address to the destination address to allow ARP resolution.)

Results

Check the results of the configuration:

```
[edit]
user@switch# show
vlans {
   pvlan100{
      vlan-id 100;
   }
}
```

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 322
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2 on page 323
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3 on page 324

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_xe-0/0/15.0__, Created at: Wed Sep 16 23:15:27 2015
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)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/15.0*,
 untagged, access

VLAN: __pvlan_pvlan100_xe-0/0/16.0__, Created at: Wed Sep 16 23:15:27 2015
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)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/16.0*,
 untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Wed Sep 16 23:15:27 2015
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)
      xe-0/0/0.0*, tagged, trunk      xe-0/0/5.0*, tagged, trunk
VLAN: default, Created at: Wed Sep 16 03:03:18 2015
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: Wed Sep 16 23:15:27 2015
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)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/11.0*,
 untagged, access
      xe-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Wed Sep 16 23:15:27 2015
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)
     xe-0/0/0.0*, tagged, trunk     xe-0/0/5.0*, tagged, trunk     xe-0/0/13.0*,
 untagged, access
      xe-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Wed Sep 16 23:15:27 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk      xe-0/0/11.0*, untagged, access
      xe-0/0/12.0*, untagged, access
      xe-0/0/13.0*, untagged, access
      xe-0/0/14.0*, untagged, access
      xe-0/0/15.0*, untagged, access
      xe-0/0/16.0*, untagged, access
Secondary VLANs: Isolated 2, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_xe-0/0/15.0__
      __pvlan_pvlan100_xe-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 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_xe-0/0/17.0__, Created at: Wed Sep 16 23:19:22 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/17.0*, untagged, access

VLAN: __pvlan_pvlan100_isiv__, Created at: Wed Sep 16 23:19:22 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk

VLAN: default, Created at: Wed Sep 16 03:03:18 2015
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: Wed Sep 16 23:19:22 2015
```

```
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/11.0*, untagged, access
      xe-0/0/12.0*, untagged, access

VLAN: hr-comm, Created at: Wed Sep 16 23:19:22 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/13.0*, untagged, access
      xe-0/0/14.0*, untagged, access

VLAN: pvlan100, Created at: Wed Sep 16 23:19:22 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/5.0*, tagged, trunk
      xe-0/0/11.0*, untagged, access
      xe-0/0/12.0*, untagged, access
      xe-0/0/13.0*, untagged, access
      xe-0/0/14.0*, untagged, access
      xe-0/0/17.0*, untagged, access
Secondary VLANs: Isolated 1, Community  2, Inter-switch-isolated  1
  Isolated VLANs :
      __pvlan_pvlan100_xe-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 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: Wed Sep 16 23:22:40 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk

VLAN: default, Created at: Wed Sep 16 03:03:18 2015
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: Wed Sep 16 23:22:40 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk

VLAN: hr-comm, Created at: Wed Sep 16 23:22:40 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, trunk

VLAN: pvlan100, Created at: Wed Sep 16 23:22:40 2015
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)
      xe-0/0/0.0*, tagged, trunk
      xe-0/0/1.0*, tagged, 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 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 226
- Using an IRB Interface in a Private VLAN on a Switch on page 451
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241
- Example: Configuring a Private VLAN on a Single QFX Series Switch on page 279

Example: Configuring a Private VLAN Spanning Multiple EX Series 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 on EX Series switches 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 EX Series switches. The example creates one primary PVLAN, containing multiple secondary VLANs:

NOTE: **Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.**

Requirements

This example uses the following hardware and software components:

- Three EX Series switches
- Junos OS Release 10.4 or later for EX Series switches

Before you begin configuring a PVLAN, make sure you have created and configured the necessary VLANs. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98.

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 EX Series switches, 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 each other even though they are included within the same domain. See "Understanding Private VLANs" on page 226.

Figure 19 on page 297 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 21: PVLAN Topology Spanning Multiple Switches

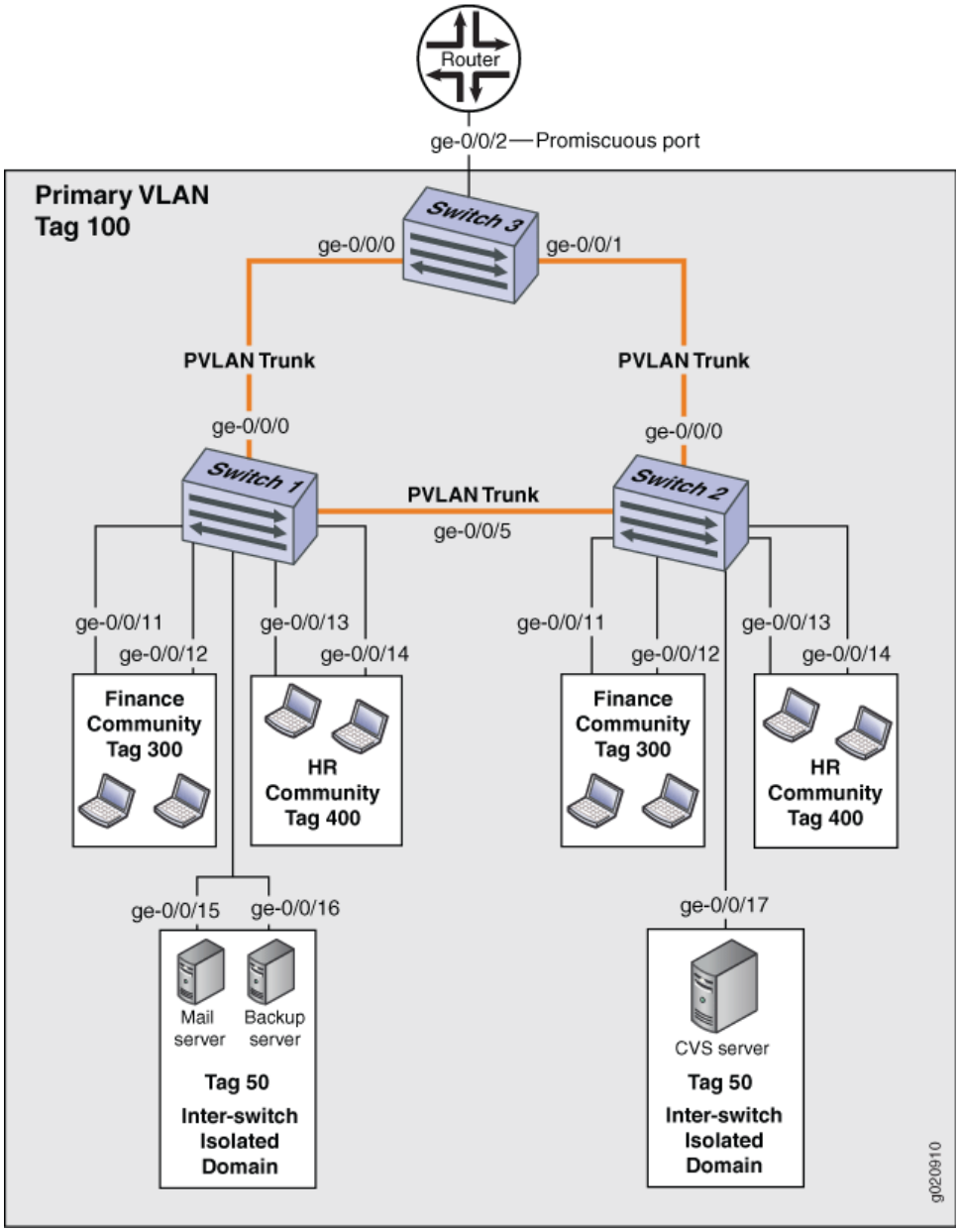


Table 55 on page 297, Table 56 on page 298, and Table 57 on page 298 list the settings for the example topology.

Table 61: Components of Switch 1 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-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
Interfaces in VLAN **isolation**	**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 62: Components of Switch 2 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-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
Interfaces in VLAN **isolation**	**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 63: Components of Switch 3 in the Topology for Configuring a PVLAN Spanning Multiple EX Series Switches

Property	Settings
VLAN names and tag IDs	**primary-vlan**, tag **100** **isolation-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

CLI Quick Configuration

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.
- Configuring a voice over IP (VoIP) VLAN on PVLAN interfaces is not supported.
- If you are going to configure a community VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- Secondary VLANs and the PVLAN trunk port must be committed on a single commit if MVRP is configured on the PVLAN trunk port.

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 no-local-switching
set vlans pvlan100 isolation-id 50
```

Step-by-Step Procedure

Complete the configuration steps below in the order shown—also, complete all steps before committing the configuration in a single commit. This is the easiest way to avoid error messages triggered by violating any of these three rules:

- If you are going to configure a community VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- If you are going to configure an isolation VLAN ID, you must first configure the primary VLAN and the PVLAN trunk port.
- Secondary vlans and a PVLAN trunk must be committed on a single commit.

To configure a PVLAN on Switch 1 that will span multiple switches:

1. Set the VLAN ID for the primary VLAN:

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

2. 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
```

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

```
[edit vlans]
user@switch# set pvlan100 no-local-switching
```

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

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

user@switch# set pvlan100 vlan–id 100
```

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

```
[edit vlans]
user@switch# set finance-comm interface interfacege-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# 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 inter-switch isolated ID to create an inter-switch isolated domain that spans the switches:

```
[edit vlans]
user@switch# set pvlan100 isolation-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;
    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;
      }
    }
    no-local-switching;
    isolation-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 inter-switch 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 no-local-switching
set vlans pvlan100 isolation-id 50
```

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# finance-comm vlan-id 300

user@switch# set pvlan100 vlan–id 100
```

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# 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 have no local switching:

```
[edit vlans]
user@switch# set pvlan100 no-local-switching
```

10. Set the inter-switch isolated ID to create an inter-switch isolated domain that spans the switches:

```
[edit vlans]
user@switch# set pvlan100 isolation-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;
        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;
        }
        no-local-switching;
        isolation-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 no-local-switching
set vlans pvlan100 isolation-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

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

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# 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 have no local switching:

   ```
   [edit vlans]
   user@switch# set pvlan100 no-local-switching
   ```

8. Set the inter-switch isolated ID to create an inter-switch isolated domain that spans the switches:

```
[edit vlans]
user@switch#  set pvlan100 isolation-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;
        }
        no-local-switching;
        isolation-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 337
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 2 on page 338
- Verifying That the Primary VLAN and Secondary VLANs Were Created on Switch 3 on page 340

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 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 is PVLAN 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 two 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

- Example: Configuring a Private VLAN on a Single EX Series Switch on page 284
- Creating a Private VLAN on a Single EX Series Switch (CLI Procedure) on page 271
- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*
- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241

Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch (CLI Procedure)

Starting with Junos OS Release 14.1X53-D10, you can configure a routed VLAN interface (RVI) for a private VLAN (PVLAN) on an EX8200 switch or EX8200 Virtual Chassis. Instead of a router connected to a promiscuous port routing Layer 3 traffic between isolated and community members, you can alternatively use an RVI.

To set up routing within a PVLAN, one RVI must be configured for the primary VLAN on one EX8200 switch or EX8200 Virtual Chassis in the PVLAN domain. This RVI serves the entire PVLAN domain regardless of whether the domain consists of one or more switches. After you configure the RVI, Layer 3 packets received by the secondary VLAN interfaces are mapped to and routed by the RVI.

When setting up the RVI, you must also enable proxy Address Resolution Protocol (ARP) so that the RVI can handle ARP requests received by the secondary VLAN interfaces.

This topic describes how to configure an RVI for a PVLAN.

Before you begin, configure the PVLAN as described in "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 271 or *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*.

To configure an RVI for a PVLAN:

1. Create a logical Layer 3 RVI on a subnet for the primary VLAN's broadcast domain:

   ```
   [edit interfaces]
   user@switch# set vlan unit logical-unit-number family inet address inet-address
   ```

2. Enable unrestricted proxy ARP on the RVI:

   ```
   [edit interfaces]
   user@switch# set vlan unit logical-unit-number proxy-arp unrestricted
   ```

3. Disable sending protocol redirect messages on the RVI:

   ```
   [edit interfaces]
   user@switch# set vlan unit logical-unit-number family inet no-redirects
   ```

4. Link the primary VLAN to the RVI:

```
[edit vlans]
user@switch#  set vlan-name l3-interface vlan.logical-unit-number
```

The value of *logical-unit-number* is the same value that you supplied for *logical-unit-number* in the previous steps.

Release History Table

Release	Description
14.1X53-D10	Starting with Junos OS Release 14.1X53-D10, you can configure a routed VLAN interface (RVI) for a private VLAN (PVLAN) on an EX8200 switch or EX8200 Virtual Chassis.

Related Documentation

- Understanding Private VLANs on page 226

Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch

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 16 on page 253 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 22 on page 343 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 22: PVLAN Topology with Secondary VLAN Trunk Ports and Promiscuous Access Port

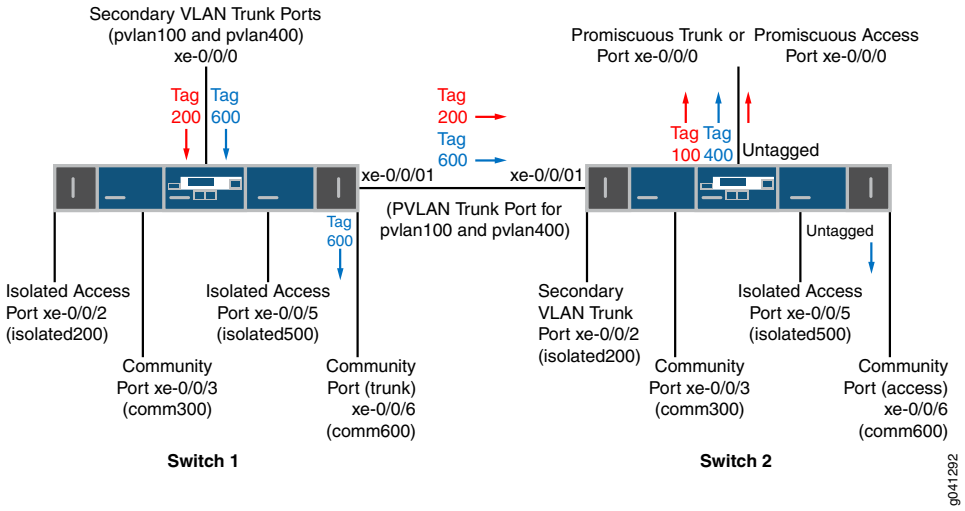


Table 64 on page 344 and Table 65 on page 344 list the settings for the example topology on both switches.

Table 64: 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
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 65: 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

Table 65: Components of the Topology for Configuring a Secondary VLAN Trunk on Switch 2 (continued)

Component	Description
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 22 on page 343 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 354
- Verifying The Ethernet Switching Table Entries on page 354

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 244
- Understanding Private VLANs on page 226

- Understanding PVLAN Traffic Flows Across Multiple Switches on page 241
- Understanding Egress Firewall Filters with PVLANs on page 253

Verifying That a Private VLAN Is Working on a Switch

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) on page 271
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277

Troubleshooting Private VLANs on QFX Switches

Use the following information to troubleshoot a private VLAN configuration.

- Limitations of Private VLANs on page 360
- Forwarding with Private VLANs on page 361
- Egress Firewall Filters with Private VLANs on page 362
- Egress Port Mirroring with Private VLANs on page 363

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.
- If you want to change a primary VLAN to be a secondary VLAN, you must first change it to a normal VLAN and commit the change. For example, you would follow this procedure:

  1. Change the primary VLAN to be a normal VLAN.
  2. Commit the configuration.
  3. Change the normal VLAN to be a secondary VLAN.
  4. Commit the configuration.

Follow the same sequence of commits if you want to change a secondary VLAN to be a primary VLAN. That is, make the secondary VLAN a normal VLAN and commit that change and then change the normal VLAN to be a primary VLAN.

Forwarding with Private VLANs

Problem **Description:**

- When isolated VLAN or community VLAN tagged traffic is received on a PVLAN trunk port, MAC addresses are learned from the primary VLAN. This means that output from the show ethernet-switching table command shows that MAC addresses are learned from the primary VLAN and replicated to secondary VLANs. This behavior has no effect on forwarding decisions.
- If a packet with a secondary VLAN tag is received on a promiscuous port, it is accepted and forwarded.
- If a packet is received on a PVLAN trunk port and meets both of the conditions listed below, it is dropped.
  - The packet has a community VLAN tag.
  - The packet is destined to a unicast MAC address or multicast group MAC address that was learned on an isolated VLAN.
- If a packet is received on a PVLAN trunk port and meets both of the conditions listed below, it is dropped.
  - The packet has an isolated VLAN tag.
  - The packet is destined to a unicast MAC address or multicast group MAC address that was learned on a community VLAN.
- If a packet with a primary VLAN tag is received by a secondary (isolated or community) VLAN port, the secondary port forwards the packet.
- If you configure a community VLAN on one device and configure another community VLAN on a second device and both community VLANs use the same VLAN ID, traffic for one of the VLANs can be forwarded to the other VLAN. For example, assume the following configuration:
  - Community VLAN comm1 on switch 1 has VLAN ID 50 and is a member of primary VLAN pvlan100.
  - Community VLAN comm2 on switch 2 also has VLAN ID 50 and is a member of primary VLAN pvlan200.
  - Primary VLAN pvlan100 exists on both switches.

  If traffic for comm1 is sent from switch 1 to switch 2, it will be sent to the ports participating in comm2. (The traffic will also be forwarded to the ports in comm1, as you would expect.)

Solution These are expected behaviors.

Egress Firewall Filters with Private VLANs

Problem **Description:** 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 that carries an isolated VLAN 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 secondary VLAN trunk port that carries a community VLAN 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).

Solution These are expected behaviors. They occur only if you apply a firewall filter to a private VLAN in the output direction and do not occur if you apply a firewall filter to a private VLAN in the input direction.

See Also
- Understanding Private VLANs on page 226

Egress Port Mirroring with Private VLANs

Problem

Description: If you create a port-mirroring configuration that mirrors private VLAN (PVLAN) traffic on egress, the mirrored traffic (the traffic that is sent to the analyzer system) has the VLAN tag of the ingress VLAN instead of the egress VLAN. For example, assume the following PVLAN configuration:

- Promiscuous trunk port that carries primary VLANs pvlan100 and pvlan400.
- Isolated access port that carries secondary VLAN isolated200. This VLAN is a member of primary VLAN pvlan100.
- Community port that carries secondary VLAN comm300. This VLAN is also a member of primary VLAN pvlan100.
- Output interface (monitor interface) that connects to the analyzer system. This interface forwards the mirrored traffic to the analyzer.

If a packet for pvlan100 enters on the promiscuous trunk port and exits on the isolated access port, the original packet is untagged on egress because it is exiting on an access port. However, the mirror copy retains the tag for pvlan100 when it is sent to the analyzer.

Here is another example: If a packet for comm300 ingresses on the community port and egresses on the promiscuous trunk port, the original packet carries the tag for pvlan100 on egress, as expected. However, the mirrored copy retains the tag for comm300 when it is sent to the analyzer.

Solution

This is expected behavior.

Related Documentation

- Understanding Private VLANs on page 226
- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 244
- Creating a Private VLAN on a Single QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch on page 342

CHAPTER 13

Configuring Routed VLAN Interfaces

Configuring Routed VLAN Interfaces on Switches (CLI Procedure)

Routed VLAN interfaces (RVIs) allow the switch to recognize packets that are being sent to local addresses so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated.

An interface named vlan functions as a logical router on which you can configure a Layer 3 logical interface for each virtual LAN (VLAN). For redundancy, you can combine an RVI with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments.

Jumbo frames of up to 9216 bytes are supported on an RVI. To route jumbo data packets on the RVI, you must configure the jumbo MTU size on the member physical interfaces of the VLAN that you have associated with the RVI, as well as on the RVI itself (the interface named vlan).

CAUTION: **Setting or deleting the jumbo MTU size on the RVI (the interface named vlan) while the switch is transmitting packets might result in dropped packets.**

To configure the RVI:

1. Create a Layer 2 VLAN by assigning it a name and a VLAN ID:

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

2. Assign an interface to the VLAN by naming the VLAN as a trunk member on the logical interface, thereby making the interface part of the VLAN's broadcast domain:

   ```
   [edit]
   ```

```
user@switch# set interfaces interface-name unit logical-unit-number family ethernet-switching
vlan members vlan-name
```

3. Create a logical Layer 3 RVI (its name will be vlan.*logical-interface-number*, where the value for *logical-interface-number* is the value you supplied for *vlan-id* in Step 1; in the following command, it is the *logical-unit-number*) on a subnet for the VLAN's broadcast domain:

```
[edit]
user@switch# set interfaces vlan unit logical-unit-number family inet address inet-address
```

4. Link the Layer 2 VLAN to the logical Layer 3 interface:

```
[edit]
user@switch# set vlans vlan-name l3-interface vlan.logical-interface-number
```

NOTE: Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple Layer 2 VLANs. Within a VLAN, traffic is switched, while across VLANs, traffic is routed.

5. (Optional) On an EX8200 switch, enable an input counter for tracking or billing purposes:

```
[edit]
user@switch# set vlans vlan-name l3-interface vlan logical-interface-number
l3-interface-ingress-counting
```

NOTE: The input counter is maintained by a firewall filter—these counters are allocated on a first-come, first-served basis.

Related Documentation

- Verifying Routed VLAN Interface Status and Statistics on EX Series Switches on page 368
- Understanding Integrated Routing and Bridging on page 445

Configuring a Routed VLAN Interface in a Private VLAN on an EX Series Switch (CLI Procedure)

Starting with Junos OS Release 14.1X53-D10, you can configure a routed VLAN interface (RVI) for a private VLAN (PVLAN) on an EX8200 switch or EX8200 Virtual Chassis. Instead of a router connected to a promiscuous port routing Layer 3 traffic between isolated and community members, you can alternatively use an RVI.

To set up routing within a PVLAN, one RVI must be configured for the primary VLAN on one EX8200 switch or EX8200 Virtual Chassis in the PVLAN domain. This RVI serves the entire PVLAN domain regardless of whether the domain consists of one or more switches. After you configure the RVI, Layer 3 packets received by the secondary VLAN interfaces are mapped to and routed by the RVI.

When setting up the RVI, you must also enable proxy Address Resolution Protocol (ARP) so that the RVI can handle ARP requests received by the secondary VLAN interfaces.

This topic describes how to configure an RVI for a PVLAN.

Before you begin, configure the PVLAN as described in "Creating a Private VLAN on a Single EX Series Switch (CLI Procedure)" on page 271 or *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*.

To configure an RVI for a PVLAN:

1. Create a logical Layer 3 RVI on a subnet for the primary VLAN's broadcast domain:

   ```
   [edit interfaces]
   user@switch# set vlan unit logical-unit-number family inet address inet-address
   ```

2. Enable unrestricted proxy ARP on the RVI:

   ```
   [edit interfaces]
   user@switch# set vlan unit logical-unit-number proxy-arp unrestricted
   ```

3. Disable sending protocol redirect messages on the RVI:

   ```
   [edit interfaces]
   user@switch# set vlan unit logical-unit-number family inet no-redirects
   ```

4. Link the primary VLAN to the RVI:

   ```
   [edit vlans]
   user@switch# set vlan-name l3-interface vlan.logical-unit-number
   ```

   The value of *logical-unit-number* is the same value that you supplied for *logical-unit-number* in the previous steps.

Release History Table

Release	Description
14.1X53-D10	Starting with Junos OS Release 14.1X53-D10, you can configure a routed VLAN interface (RVI) for a private VLAN (PVLAN) on an EX8200 switch or EX8200 Virtual Chassis.

Related Documentation

- Understanding Private VLANs on page 226

Verifying Routed VLAN Interface Status and Statistics on EX Series Switches

Purpose Determine status information and traffic statistics for routed VLAN interfaces (RVIs) by using the following commands:

Action Display RVI interfaces and their current states:

```
user@switch> show interfaces vlan terse
Interface               Admin Link Proto    Local                 Remote
vlan                    up    up
vlan.111                up    up   inet     111.111.111.1/24
```

Display Layer 2 VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs:

```
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*
```

Display Ethernet switching table entries for the VLAN that is attached to the RVI:

```
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
```

Display an RVI's ingress-counting statistics with either the **show interfaces vlan detail** command or the **show interfaces vlan extensive** command. Ingress counting is displayed as **Input bytes** and **Input packets** under **Transit Statistics**.

```
user@switch> show interfaces vlan.100 detail

Logical interface vlan.100 (Index 65) (SNMP ifIndex 503) (HW Token 100) (Generation
```

```
 131)
Flags: SNMP-Traps 0x4000  Encapsulation: ENET2
Traffic statistics:
 Input bytes:    17516756
 Output bytes:   411764
 Input packets:  271745
 Output packets: 8256
Local statistics:
 Input bytes:    3240
 Output bytes:   411764
 Input packets:  54
 Output packets: 8256
Transit statistics:
 Input bytes:    17513516   0 bps
 Output bytes:   0    0 bps
 Input packets:  271745  0 pps
 Output packets: 0    0 pps
Protocol inet, Generation: 148, Route table: 0
 Flags: None
 Addresses, Flags: iS-Preferred  Is-Primary
  Destination: 50.1.1/24, Local: 50.1.1.1, Broadcast: 50.1.1.255, Generation: 136
```

Meaning

- **show interfaces vlan** displays a list of interfaces, including RVI interfaces, and their current states (up, down).
- **show vlans** displays a list of VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs.
- **show ethernet-switching table** displays the Ethernet switching table entries, including VLANs attached to the RVI.
- **show interfaces vlan detail** displays RVI ingress counting as Input Bytes and Input Packets under Transit Statistics.

Related Documentation

- Configuring Routed VLAN Interfaces on Switches (CLI Procedure) on page 365

CHAPTER 14

Configuring VLANS and VPLS Routing Instances

Guidelines for Configuring VLAN Identifiers for VLANs and VPLS Routing Instances

For a VLAN that is performing Layer 2 switching only, you do not have to specify a VLAN identifier.

For a VLAN that is performing Layer 3 IP routing, you must specify either a VLAN identifier or dual VLAN identifier tags.

For a VPLS routing instance, you must specify either a VLAN identifier or dual VLAN identifier tags.

Related Documentation

- Layer 2 Learning and Forwarding for VLANs Overview on page 27

Configuring VLAN Identifiers for VLANs and VPLS Routing Instances

You can configure VLAN identifiers for a VLAN or a VPLS routing instance in the following ways:

- By using either the **vlan-id** statement or the **vlan-tags** statement to configure a normalizing VLAN identifier. This topic describes how normalizing VLAN identifiers are processed and translated in a VLAN or a VPLS routing instance.
- By using the **input-vlan-map** and the **output-vlan-map** statements at the **[edit interfaces *interface-name* unit *logic-unit-number*]** or **[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logic-unit-number*]** hierarchy level to configure VLAN mapping.

The **vlan-id** and **vlan-tags** statements are used to specify the normalizing VLAN identifier under the VLAN or VPLS routing instance. The normalizing VLAN identifier is used to perform the following functions:

- Translate, or normalize, the VLAN tags of packets received into a learn VLAN identifier.
- Create multiple learning domains that each contain a learn VLAN identifier. A learning domain is a MAC address database to which MAC addresses are added based on the learn VLAN identifier.

NOTE: You cannot configure VLAN mapping using the **input-vlan-map** and **output-vlan-map** statements if you configure a normalizing VLAN identifier for a VLAN or VPLS routing instance using the **vlan-id** or **vlan-tags** statements.

To configure a VLAN identifier for a VLAN, include either the **vlan-id** or the **vlan-tags** statement at the **[edit interfaces** *interface-name* **unit** *logic-unit-number***]** or **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logic-unit-number***]** hierarchy level, and then include that logical interface in the VLAN configuration.

For a VPLS routing instance, include either the **vlan-id** or **vlan-tags** statement at the **[edit interfaces** *interface-name* **unit** *logic-unit-number***]** or **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logic-unit-number***]** hierarchy level, and then include that logical interface in the VPLS routing instance configuration.

NOTE: ACX Series routers do not support the **[edit logical-systems]** hierarchy.

NOTE: For a single VLAN or VPLS routing instance, you can include either the **vlan-id** or the **vlan-tags** statement, but not both. If you do not configure a **vlan-id**, **vlan-tags**, or **vlan-id-list [** *vlan-id-numbers* **]** for the VLAN or the VPLS routing instance, the Layer 2 packets received are forwarded to the outbound Layer 2 interface without having the VLAN tag modified unless an **output-vlan-map** is configured on the Layer 2 interface. This results in a frame being forwarded to a Layer 2 interface with a VLAN tag that is different from what is configured for the Layer 2 interface. Note that a frame received from the Layer 2 interface is still required to match the VLAN tag(s) specified in the interface configuration. The invalid configuration may cause a Layer 2 loop to occur.

The VLAN tags associated with the inbound logical interface are compared with the normalizing VLAN identifier. If the tags are different, they are rewritten as described in Table 66 on page 375. The source MAC address of a received packet is learned based on the normalizing VLAN identifier.

NOTE: You do not have to specify a VLAN identifier for a VLAN that is performing Layer 2 switching only. To support Layer 3 IP routing, you must specify either a VLAN identifier or a pair of VLAN tags. However, you cannot specify the same VLAN identifier for more than one VLAN within a routing instance. Each VLAN must have a unique VLAN identifier.

If the VLAN tags associated with the outbound logical interface and the normalizing VLAN identifier are different, the normalizing VLAN identifier is rewritten to match the VLAN tags of the outbound logical interface, as described in Table 67 on page 376.

For the packets sent over the VPLS routing instance to be tagged by the normalizing VLAN identifier, include one of the following configuration statements:

- **vlan-id** *number* to tag all packets that are sent over the VPLS virtual tunnel (VT) interfaces with the VLAN identifier.
- **vlan-tags outer** *number* **inner** *number* to tag all packets sent over the VPLS VT interfaces with dual outer and inner VLAN tags.

Use the **vlan-id none** statement to have the VLAN tags removed from packets associated with an inbound logical interface when those packets are sent over VPLS VT interfaces. Note that those packets might still be sent with other customer VLAN tags.

The **vlan-id all** statement enables you to configure bridging for several VLANs with a minimum amount of configuration. Configuring this statement creates a learning domain for:

- Each inner VLAN, or learn VLAN, identifier of a logical interface configured with two VLAN tags
- Each VLAN, or learn VLAN, identifier of a logical interface configured with one VLAN tag

We recommend that you do not use customer VLAN IDs in a VPLS routing instance because customer VLAN IDs are used for learning only.

You should use the service VLAN ID in a VPLS routing instance, as in the following configuration:

```
[edit]
interface ge-1/1/1 {
   vlan-tagging;
   unit 1 {
      vlan-id s1; /* Service vlan */
      encapsulation vlan-vpls;
      input-vlan-map pop; /* Pop the service vlan on input */
      output-vlan-map push; /* Push the service vlan on output */
   }
}
interface ge-1/1/2 {
   encapsulation ethernet-vpls;
   unit 0;
```

```
}
routing-instances {
   V1 {
      instance-type vpls;
      vlan-id all;
      interface ge-1/1/1.1;
      interface ge-1/1/2.0;
   }
}
```

NOTE: If you configure the **vlan-id all** statement in a VPLS routing instance, we recommend using the **input-vlan-map pop** and **output-vlan-map push** statements on the logical interface to pop the service VLAN ID on input and push the service VLAN ID on output and in this way, limit the impact of double-tagged frames on scaling. You cannot use the native **vlan-id** statement when the **vlan-id all** statement is included in the configuration.

The **vlan-id-list [** ***vlan-id-numbers*** **]** statement enables you to configure bridging for multiple VLANs on a trunk interface. Configuring this statement creates a learning domain for:

- Each VLAN listed: **vlan-id-list [100 200 300]**
- Each VLAN in a range: **vlan-id-list [100-200]**
- Each VLAN in a list and range combination: **vlan-id-list [50, 100-200, 300]**

The following steps outline the process for bridging a packet received over a Layer 2 logical interface when you specify a normalizing VLAN identifier using either the **vlan-id** ***number*** or **vlan-tags** statement for a VLAN or a VPLS routing instance:

1. When a packet is received on a physical port, it is accepted only if the VLAN identifier of the packet matches the VLAN identifier of one of the logical interfaces configured on that port.

2. The VLAN tags of the received packet are then compared with the normalizing VLAN identifier. If the VLAN tags of the packet are different from the normalizing VLAN identifier, the VLAN tags are rewritten as described in Table 66 on page 375.

3. If the source MAC address of the received packet is not present in the source MAC table, it is learned based on the normalizing VLAN identifier.

4. The packet is then forwarded toward one or more outbound Layer 2 logical interfaces based on the destination MAC address. A packet with a known unicast destination MAC address is forwarded only to one outbound logical interface. For each outbound Layer 2 logical interface, the normalizing VLAN identifier configured for the VLAN or VPLS routing instance is compared with the VLAN tags configured on that logical interface. If the VLAN tags associated with an outbound logical interface do not match the normalizing VLAN identifier configured for the VLAN or VPLS routing instance, the VLAN tags are rewritten as described in Table 67 on page 376.

The tables below show how VLAN tags are applied for traffic sent to and from the VLAN, depending on how the **vlan-id** and **vlan-tags** statements are configured for the VLAN and on how identifiers are configured for the logical interfaces in a VLAN or VPLS routing instance. Depending on your configuration, the following rewrite operations are performed on VLAN tags:

- **pop**—Remove a VLAN tag from the top of the VLAN tag stack.
- **pop-pop**—Remove both the outer and inner VLAN tags of the frame.
- **pop-swap**—Remove the outer VLAN tag of the frame and replace the inner VLAN tag of the frame.
- **swap**—Replace the VLAN tag of the frame.
- **push**—Add a new VLAN tag to the top of the VLAN stack.
- **push-push**—Push two VLAN tags in front of the frame.
- **swap-push**—Replace the VLAN tag of the frame and add a new VLAN tag to the top of the VLAN stack.
- **swap-swap**—Replace both the outer and inner VLAN tags of the frame.

Table 66 on page 375 shows specific examples of how the VLAN tags for packets sent to the VLAN are processed and translated, depending on your configuration. "–" means that the statement is not supported for the specified logical interface VLAN identifier. "No operation" means that the VLAN tags of the received packet are not translated for the specified input logical interface.

Table 66: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a VLAN

VLAN Identifier of Logical Interface	VLAN Configurations for a VLAN			
	vlan-id none	vlan-id 200	vlan-id all	vlan tags outer 100 inner 300
none	No operation	push 200	–	push 100, push 300
200	pop 200	No operation	No operation	swap 200 to 300, push 100
1000	pop 1000	swap 1000 to 200	No operation	swap 1000 to 300, push 100
vlan-tags outer 2000 inner 300	pop 2000, pop 300	pop 2000, swap 300 to 200	pop 2000	swap 2000 to 100
vlan-tags outer 100 inner 400	pop 100, pop 400	pop 100, swap 400 to 200	pop 100	swap 400 to 300
vlan-id-range 10-100	–	–	No operation	–
vlan-tags outer 200 inner-range 10-100	–	–	pop 200	–

Table 67 on page 376 shows specific examples of how the VLAN tags for packets sent from the VLAN are processed and translated, depending on your configuration. "–" means that the statement is not supported for the specified logical interface VLAN identifier. "No operation" means that the VLAN tags of the outbound packet are not translated for the specified output logical interface.

Table 67: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a VLAN

VLAN Identifier of Logical Interface	VLAN Configurations for a VLAN			
	vlan-id none	vlan-id 200	vlan-id all	vlan tags outer 100 inner 300
none	no operation	pop 200	–	pop 100, pop 300
200	push 200	No operation	No operation	pop 100, swap 300 to 200
1000	push 1000	swap 200 to 1000	No operation	pop 100, swap 300 to 1000
vlan-tags outer 2000 inner 300	push 2000, push 300	swap 200 to 300, push 2000	push 2000	swap 100 to 2000
vlan-tags outer 100 inner 400	push 100, push 400	swap 200 to 400, push 100	push 100	swap 300 to 400
vlan-id-range 10-100	–	–	No operation	–
vlan-tags outer 200 inner-range 10-100	–	–	push 200	–

CHAPTER 15

Configuring VLANs in Transparent Mode on Security Devices

Layer 2 Transparent Mode Overview

A device operates in transparent mode when there are interfaces defined as Layer 2 interfaces. The device operates in route mode (the default mode) if there are no physical interfaces configured as Layer 2 interfaces.

For SRX Series devices, transparent mode provides full security services for Layer 2 switching capabilities. On these SRX Series devices, you can configure one or more VLANs to perform Layer 2 switching. A VLAN is a set of logical interfaces that share the same flooding or broadcast characteristics. Like a virtual LAN (VLAN), a VLAN spans one or more ports of multiple devices. Thus, the SRX Series device can function as a Layer 2 switch with multiple VLANs that participate in the same Layer 2 network.

In transparent mode, the SRX Series device filters packets that traverse the device without modifying any of the source or destination information in the IP packet headers. Transparent mode is useful for protecting servers that mainly receive traffic from untrusted sources because there is no need to reconfigure the IP settings of routers or protected servers.

In transparent mode, all physical ports on the device are assigned to Layer 2 interfaces. Do not route Layer 3 traffic through the device. Layer 2 zones can be configured to host Layer 2 interfaces, and security policies can be defined between Layer 2 zones. When packets travel between Layer 2 zones, security policies can be enforced on these packets.

Table 68 on page 378 lists the security features that are supported and are not supported in transparent mode for Layer 2 switching.

Table 68: Security Features Supported in Transparent Mode

Mode Type	Supported	Not Supported
Transparent mode	• Application Layer Gateways (ALGs) • Firewall User Authentication (FWAUTH) • Intrusion Detection and Prevention (IDP) • Screen • AppSecure • Unified Threat Management (UTM)	• Network Address Translation (NAT) • VPN

NOTE:

- Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, mixed mode is the default mode, and you can configure an SRX Series device using both transparent mode (Layer 2) and route mode (Layer 3) simultaneously, with no reboot required.
- On all SRX Series devices, transparent mode is not supported on mPIMs.
- On SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the DHCP server propagation is not supported in Layer 2 transparent mode.

Layer 2 Switching Exceptions on SRX Series Devices

The switching functions on the SRX Series devices are similar to the switching features on Juniper Networks MX Series routers. However, the following Layer 2 networking features on MX Series routers are not supported on SRX Series devices:

- Layer 2 control protocols—These protocols are used on MX Series routers for Rapid Spanning Tree Protocol (RSTP) or Multiple Spanning Tree Protocol (MSTP) in customer edge interfaces of a VPLS routing instance.
- Virtual switch routing instance—The virtual switching routing instance is used on MX Series routers to group one or more VLANs.
- Virtual private LAN services (VPLS) routing instance—The VPLS routing instance is used on MX Series routers for point-to-multipoint LAN implementations between a set of sites in a VPN.

In addition, the SRX Series devices do not support the following Layer 2 features:

- Spanning Tree Protocol (STP), RSTP, or MSTP—It is the user's responsibility to ensure that no flooding loops exist in the network topology.
- Internet Group Management Protocol (IGMP) snooping—Host-to-router signaling protocol for IPv4 used to report their multicast group memberships to neighboring routers and determine whether group members are present during IP multicasting.

- Double-tagged VLANs or IEEE 802.1Q VLAN identifiers encapsulated within 802.1Q packets (also called “Q in Q” VLAN tagging)—Only untagged or single-tagged VLAN identifiers are supported on SRX Series devices.
- Nonqualified VLAN learning, where only the MAC address is used for learning within the VLAN—VLAN learning on SRX Series devices is qualified; that is, both the VLAN identifier and MAC address are used.

Also, on SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX550, or SRX650 devices, some features are not supported. (Platform support depends on the Junos OS release in your installation.) The following features are not supported for Layer 2 transparent mode on the mentioned devices:

- G-ARP on the Layer 2 interface
- IP address monitoring on any interface
- Transit traffic through IRB
- IRB interface in a routing instance
- IRB interface handling of Layer 3 traffic

NOTE: The IRB interface is a pseudointerface and does not belong to the reth interface and redundancy group.

Layer 2 Transparent Mode on the SRX5000 Line Module Port Concentrator

The SRX5000 line Module Port Concentrator (SRX5K-MPC) supports Layer 2 transparent mode and processes the traffic when the SRX Series device is configured in Layer 2 transparent mode.

When the SRX5K-MPC is operating in Layer 2 mode, you can configure all interfaces on the SRX5K-MPC as Layer 2 switching ports to support Layer 2 traffic.

The security processing unit (SPU) supports all security services for Layer 2 switching functions, and the MPC delivers the ingress packets to the SPU and forwards the egress packets that are encapsulated by the SPU to the outgoing interfaces.

When the SRX Series device is configured in Layer 2 transparent mode, you can enable the interfaces on the MPC to work in Layer 2 mode by defining one or more logical units on a physical interface with the family address type as **Ethernet switching**. Later you can proceed with configuring Layer 2 security zones and configuring security policies in transparent mode. Once this is done, next-hop topologies are set up to process ingress and egress packets.

Configuring Out-of-Band Management on SRX Devices

You can configure the **fxp0** out-of-band management interface on the SRX Series device as a Layer 3 interface, even if Layer 2 interfaces are defined on the device. With the exception of the **fxp0** interface, you can define Layer 2 and Layer 3 interfaces on the device’s network ports.

NOTE: There is no fxp0 out-of-band management interface on the SRX100. SRX210, SRX220. SRX240. SRX300, SRX320, SRX340, SRX345 or SRX650 devices. (Platform support depends on the Junos OS release in your installation.)

Related Documentation

- Understanding VLANs on Security Devices on page 380
- Layer 2 Transparent Mode Overview on page 377
- Understanding Layer 2 Interfaces on Security Devices on page 637
- Understanding Layer 2 Security Zones on page 651
- Understanding Security Policies in Transparent Mode on page 653
- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Layer 2 Logical Interfaces on Security Devices on page 638
- Understanding Layer 2 Interfaces on Security Devices on page 637
- Understanding Mixed Mode (Transparent and Route Mode) on Security Devices on page 639

Understanding VLANs on Security Devices

The packets that are forwarded within a VLAN are determined by the VLAN ID of the packets and the VLAN ID of the VLAN. Only the packets with VLAN IDs that match the VLAN ID configured for a VLAN are forwarded within the VLAN.

When configuring VLANs, you can specify either a single VLAN ID or a list of specific VLAN IDs. If you specify a list of VLAN IDs, a VLAN is created for each VLAN ID in the list. Certain VLAN properties, such as the integrated routing and bridging interface (IRB), are not configurable if VLANs are created in this manner.

Each Layer 2 logical interface configured on the device is implicitly assigned to a VLAN based on the VLAN ID of the packets accepted by the interface. You do not need to explicitly define the logical interfaces when configuring a VLAN.

You can configure one or more static MAC addresses for a logical interface in a VLAN; this is only applicable if you specified a single VLAN ID when creating the VLAN.

NOTE: If a static MAC address you configure for a logical interface appears on a different logical interface, packets sent to that interface are dropped.

You can configure the following properties that apply to all VLANs on the SRX Series device:

- Layer 2 address learning—Layer 2 address learning is enabled by default. A VLAN learns unicast media access control (MAC) addresses to avoid flooding packets to all

interfaces in the VLAN. Each VLAN creates a source MAC entry in its forwarding tables for each source MAC address learned from packets received on interfaces that belong to the VLAN. When you disable MAC learning, source MAC addresses are not dynamically learned, and any packets sent to these source addresses are flooded into a VLAN.

- Maximum number of MAC addresses learned from all logical interfaces on the SRX Series device—After the MAC address limit is reached, the default is for any incoming packets with a new source MAC address to be forwarded. You can specify that the packets be dropped instead. The default limits of MAC addresses for the SRX Series devices are shown in Table 69 on page 381 and Table 70 on page 381. (Platform support depends on the Junos OS release in your installation.)

Table 69: MAC Addresses Default Limits for Junos OS Release 15.1X49-D30 and Earlier

SRX Series Devices	Default Limit for MAC Addresses
SRX100 SRX210	1024
SRX220	2048
SRX240	4096
SRX650	16,384
SRX3400 SRX3600 SRX5600 SRX5800	131,071

Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, default limits for MAC addresses are more uniform.

Table 70: MAC Addresses Default Limits for Junos OS Release Starting in Junos OS 15.1X49-D40

SRX Series Devices	Default Limit for MAC Addresses
SRX300 SRX320 SRX340 SRX345	16,383
SRX1500	24,575
SRX4100	65536

Table 70: MAC Addresses Default Limits for Junos OS Release Starting in Junos OS 15.1X49-D40 (continued)

SRX Series Devices	Default Limit for MAC Addresses
SRX4200	65536
SRX4600	65536
SRX5600 SRX5800	131,071

- Timeout interval for MAC table entries. By default, the timeout interval for MAC table entries is 300 seconds. The minimum you can configure is 10 seconds and the maximum is 64,000 seconds. The timeout interval applies only to dynamically learned MAC addresses. This value does not apply to configured static MAC addresses, which never time out.

Release History Table

Release	Description
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, default limits for MAC addresses are more uniform.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring VLANs on Security Devices on page 382
- Understanding Integrated Routing and Bridging on page 445
- Understanding Layer 2 Interfaces on Security Devices on page 637
- Layer 2 Learning and Forwarding for VLANs Overview on page 27

Example: Configuring VLANs on Security Devices

This example shows how to configure VLANs.

NOTE: Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, new terminology and CLI keywords are used for switching functions. If your installation uses a Junos OS release preceding 15.1X49-D10, see "Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices" on page 385 to determine how you must modify configuration tasks for implementation in earlier Junos OS environments.

- Requirements on page 383
- Overview on page 383

Requirements

Before you begin, determine the properties you want to configure for the VLAN. See "Understanding VLANs on Security Devices" on page 380.

Overview

In this example, you configure VLAN vlan-a for VLANs 1 and 10, and VLAN vlan-b for VLAN 2. You then limit the number of MAC addresses learned on all logical interfaces on the device to 64,000. When this limit is reached, incoming packets with a new source MAC address will be dropped.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-a vlan members 1-10
set vlans vlan-b vlan-id 2
set protocols l2-learning global-mac-limit 64000 packet-action drop
```

Step-by-Step Procedure

To configure VLANs:

1. Configure the domain type and VLANs.

   ```
   [edit]
   user@host# set vlans vlan-a vlan members 1-10
   user@host# set vlans vlan-b vlan-id 2
   ```

2. Limit the number of MAC addresses.

   ```
   [edit]
   user@host# set protocols l2-learning global-mac-limit 64000 packet-action drop
   ```

3. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Verification

To verify the configuration is working properly, enter the **show vlans** and **show protocols l2-learning** commands.

Release History Table

Release	Description
15.1X49-D10	Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, new terminology and CLI keywords are used for switching functions.

Related Documentation

- Understanding Integrated Routing and Bridging on page 445
- Understanding Layer 2 Interfaces on Security Devices on page 637
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- Understanding VLANs on Security Devices on page 380

Example: Configuring VLAN Retagging for Layer 2 Transparent Mode on a Security Device

This example shows how to configure VLAN retagging on a Layer 2 trunk interface to selectively screen incoming packets and redirect them to a security device without affecting other VLAN traffic.

Requirements

Before you begin, determine the mapping you want to include for the VLAN retagging. See "Understanding VLAN Retagging on Security Devices" on page 773.

Overview

In this example, you create a Layer 2 trunk interface called ge-3/0/0 and configure it to receive packets with VLAN identifiers 1 through 10. Packets that arrive on the interface with VLAN identifier 11 are retagged with VLAN identifier 2. Before exiting the trunk interface, VLAN identifier 2 in the retagged packets is replaced with VLAN identifier 11. All VLAN identifiers in the retagged packets change back when you exit the trunk interface.

Configuration

Step-by-Step Procedure

To configure VLAN retagging on a Layer 2 trunk interface:

1. Create a Layer 2 trunk interface.

```
[edit]
user@host#set interfaces ge-3/0/0 unit 0 family ethernet-switching interface-mode
   trunk vlan members 1–10
```

2. Configure VLAN retagging.

```
[edit]
user@host#set interfaces ge-3/0/0 unit 0 family ethernet-switching vlan-rewrite
  translate 11 2
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interfaces ge-3/0/0** command.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Layer 2 Logical Interfaces on Security Devices on page 638

Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices

Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, some Layer 2 CLI configuration statements are enhanced, and some commands are changed. Table 71 on page 385 and Table 72 on page 386 provide lists of existing commands that have been moved to new hierarchies or changed on SRX Series devices as part of this CLI enhancement effort. The tables are provided as a high-level reference only. For detailed information about these commands, see CLI Explorer.

Table 71: Enhanced Layer 2 Configuration Statement Changes

Original Hierarchy	Changed Hierarchy	Hierarchy Level	Change Description

bridge-domains
  bridge-domain--name {
  ...
  }
}
	vlans *vlans-name* {
	  ...
	  }
	}
		[edit]	Hierarchy renamed.

bridge-domains
  bridge-domain--name {
  vlan-id-list [*vlan-id*] ;
}
	vlans *vlans-name* {
	  vlan members [*vlan-id*] ;
	}
		[edit vlans *vlans-name*]	Statement renamed.

Table 71: Enhanced Layer 2 Configuration Statement Changes (continued)

Original Hierarchy	Changed Hierarchy	Hierarchy Level	Change Description

bridge-options {
interface *interface-name* {
encapsulation-type;
ignore-encapsulation-mismatch;
pseudowire-status-tlv;
static-mac *mac-address* {
vlan-id *vlan-id*;
}
}
mac-table-aging-time
seconds;
mac-table-size {
number;
packet-action drop;
}
}
	switch-options {
	interface *interface-name* {
	encapsulation-type;
	ignore-encapsulation-mismatch;
	pseudowire-status-tlv;
	static-mac *mac-address* {
	vlan-id *vlan-id*;
	}
	}
	mac-table-aging-time
	seconds;
	mac-table-size {
	number;
	packet-action drop;
	}
	}
		[edit vlans *vlans-name*]	Statement renamed.

bridge {
block-non-ip-all;
bpdu-vlan-flooding;
bypass-non-ip-unicast;
no-packet-flooding {
no-trace-route;
}
}
	ethernet-switching {
	block-non-ip-all;
	bpdu-vlan-flooding;
	bypass-non-ip-unicast;
	no-packet-flooding {
	no-trace-route;
	}
	}
		[edit security flow]	Statement renamed.

family {
bridge {
bridge-domain-type (svlan|
bvlan);
...
	family {
	ethernet-switching {
	...
		[edit interfaces *interface-name*]
		unit *unit-number*
			Hierarchy renamed.

... **routing-interface** irb.0; ...	... **l3-interface** irb.0; ...	[edit vlans *vlans-name*]	Statement renamed.

Table 72: Enhanced Layer 2 Operational Command Changes

Original Operational Command	Modified Operational Command
clear bridge mac-table	clear ethernet-switching table
clear bridge mac-table persistent-learning	clear ethernet-switching table persistent-learning
show bridge domain	show vlans
show bridge mac-table	show ethernet-switching table
show l2-learning interface	show ethernet-switching interface

NOTE: There is no fxp0 out-of-band management interface on the SRX300, SRX320, and SRX500HM devices. (Platform support depends on the Junos OS release in your installation.)

Release History Table

Release	Description
15.1X49-D10	Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, some Layer 2 CLI configuration statements are enhanced, and some commands are changed.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Understanding Switching Modes on Security Devices on page 705

CHAPTER 16

Configuring Layer 2 Protocol Tunneling

Understanding Layer 2 Protocol Tunneling on EX Series Switches

Layer 2 protocol tunneling (L2PT) enables service providers to send Layer 2 protocol data units (PDUs) across the provider's cloud and deliver them to Juniper Networks EX Series Ethernet Switches that are not part of the local broadcast domain. This feature is useful when you want to run Layer 2 protocols on a network that includes switches located at remote sites that are connected across a service provider network.

You can also use L2PT to tunnel protocols between two locally-connected user-to-network interfaces (UNIs) in the same broadcast domain, but in that case, the protocol packets are flooded in the VLAN instead of being rewritten with the tunnel MAC address.

See Feature Explorer for the list of switches that support L2PT.

This topic includes:

Benefits of Layer 2 Protocol Tunneling

- Enables you to run Layer 2 protocols on a network that includes switches located at remote sites that are connected across a service provider network.

Layer 2 Protocols Supported by L2PT on EX Series Switches

Table 73 on page 390 lists the Layer 2 protocols supported by L2PT on EX Series switches. For details on the configuration options to enable tunneling the supported protocols on each type of EX Series switch, and the releases in which those options are supported, see either of the following configuration statements:

- For non-ELS (EX2200, EX3300, EX4200, EX4500, and EX4450 switches), see the layer2-protocol-tunneling statement in the **[edit vlans *vlan-name* dot1q-tunneling]** hierarchy.
- For ELS (EX2300, EX3400, EX4300, EX4600, and EX9200 switches), see the protocol statement in the **[edit protocols layer2-control mac-rewrite interface *interface-name*]** hierarchy.

Table 73: L2PT Protocols Supported on EX Series Switches

Layer 2 Protocol That Can Be Tunneled	EX Series Switch Support
802.1X authentication	Non-ELS and ELS EX Series (except EX2300 multigigabit models)
802.3ah Operation, Administration, and Maintenance (OAM) link fault management (LFM) NOTE: If you enable L2PT for untagged OAM LFM packets, do not configure LFM on the corresponding access interface.	Non-ELS and ELS EX Series
Cisco Discovery Protocol (CDP)	Non-ELS and ELS EX Series
Ethernet local management interface (E-LMI)	Non-ELS and ELS EX Series (except EX2300 multigigabit models)
Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP)	ELS EX Series switches only
Link Aggregation Control Protocol (LACP) NOTE: If you enable L2PT for untagged LACP packets, do not configure Link Aggregation Control Protocol (LACP) on the corresponding access interface.	Non-ELS and ELS EX Series
Link Layer Discovery Protocol (LLDP)	Non-ELS and ELS EX Series
Multiple MAC Registration Protocol (MMRP)—All non-ELS EX Series switches and for ELS switches, EX4300, EX4600, and EX9200 switches only	Non-ELS and ELS EX Series (except EX2300 multigigabit models)

Table 73: L2PT Protocols Supported on EX Series Switches (continued)

Layer 2 Protocol That Can Be Tunneled	EX Series Switch Support
MVRP VLAN Registration Protocol (MVRP)	Non-ELS and ELS EX Series
Per-VLAN Spanning Tree and Per-VLAN Spanning Tree Plus (PVST+) Protocols	EX9200 switches only
Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP)	Non-ELS and ELS EX Series
Unidirectional Link Detection (UDLD)	Non-ELS and ELS EX Series (except EX2300 multigigabit models)
VLAN Spanning Tree Protocol (VSTP) NOTE: EX9200 switches do not have a separate option to enable VSTP. The L2PT configuration statement option for ELS EX Series switches that enables tunneling PVST and PVST+ (**pvstp**) also enables tunneling VSTP.	Non-ELS and ELS EX Series
VLAN Trunking Protocol (VTP)	Non-ELS and ELS EX Series

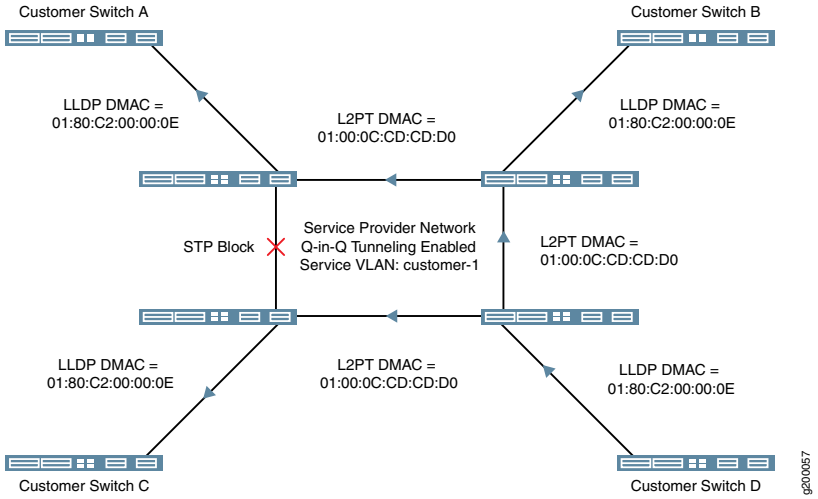
NOTE: CDP, UDLD, and VTP cannot be configured on EX Series switches. L2PT does, however, tunnel CDP, UDLD, and VTP PDUs.

How L2PT Works

L2PT works by encapsulating Layer 2 PDUs, tunneling them across a service provider network, and decapsulating them for delivery to their destination switches. L2PT encapsulates Layer 2 PDUs by enabling the ingress provider edge (PE) device to rewrite the PDUs' destination media access control (MAC) addresses before forwarding them onto the service provider network. The devices in the service provider network treat these encapsulated PDUs as multicast Ethernet packets. Upon receipt of these PDUs, the egress PE devices decapsulate them by replacing the destination MAC addresses with the address of the Layer 2 protocol that is being tunneled before forwarding the PDUs to their destination switches.

This process is illustrated in the following example for tunneling LLDP packets in Figure 23 on page 392.

Figure 23: L2PT Example



1. Customer Switch D sends an LLDP PDU to the service provider network that is ultimately intended for the other switches in the customer network.

2. The receiving provider switch rewrites the DMAC with the L2PT DMAC and sends the frame with the encapsulated LLDP PDU to the other switches in the service provider network.

3. When the other service provider switches receive the frame, they restore the LLDP DMAC and send it to Customer Switches A, B, and C.

The destination switches identify the tunneled Layer 2 control protocol by the encapsulated MAC address. The destination MAC addresses used by different protocols are listed in Table 74 on page 392:

Table 74: Protocol Destination MAC Addresses

Protocol	Ethernet Encapsulation	MAC Address
802.1X	Ether-II	01:80:C2:00:00:03
802.3ah	Ether-II	01:80:C2:00:00:02
CDP	LLC/SNAP	01:00:0C:CC:CC:CC
E-LMI	Ether-II	01:80:C2:00:00:07
GVRP	LLC/SNAP	01:80:C2:00:00:21

Table 74: Protocol Destination MAC Addresses (continued)

Protocol	Ethernet Encapsulation	MAC Address
LACP	Ether-II	01:80:C2:00:00:02
LLDP	Ether-II	01:80:C2:00:00:0E
MMRP	Ether-II	01:80:C2:00:00:20
MVRP	Ether-II	01:80:C2:00:00:21
PVSTP	LLC/SNAP	01:00:0C:CC:CC:CD
STP, RSTP, MSTP	LLC/SNAP	01:80:C2:00:00:00
UDLD	LLC/SNAP	01:00:0C:CC:CC:CC
VSTP	LLC/SNAP	01:00:0C:CC:CC:CD
VTP	LLC/SNAP	01:00:0C:CC:CC:CC

When a PE device receives a Layer 2 control PDU from any of the customer PE devices, it changes the destination MAC address to 01:00:0C:CD:CD:D0. The modified packet is then sent to the provider network. All devices on the provider network treat these packets as multicast Ethernet packets and deliver them to all PE devices for the customer. The egress PE devices receive all the control PDUs with the same MAC address (01:00:0C:CD:CD:D0). Then they identify the packet type by doing deeper packet inspection and replace the destination MAC address 01:00:0C:CD:CD:D0 with the appropriate destination address. The modified PDUs are sent out to the customer PE devices, thus ensuring the Layer 2 control PDUs are delivered, in their original state, across the provider network. The L2PT protocol is valid for all types of packets (untagged, tagged, and Q-in-Q tagged).

L2PT Basics on EX Series Switches

L2PT is enabled on a per-VLAN basis. When you enable L2PT for a particular Layer 2 protocol on a VLAN, all access interfaces are considered to be customer-facing interfaces, all trunk interfaces are considered to be service provider network-facing interfaces, and you cannot configure the specified protocol on the access interfaces. L2PT only acts on logical interfaces of the family **ethernet-switching**. L2PT PDUs are flooded to all trunk and access ports within a given S-VLAN.

NOTE: Access interfaces in an L2PT-enabled VLAN should not receive L2PT-tunneled PDUs. If an access interface does receive L2PT-tunneled PDUs, it might mean that there is a loop in the network. As a result, the interface will be shut down.

You must configure and enable Q-in-Q tunneling (802.1Q VLAN encapsulation) before you can configure L2PT. For information about Q-in-Q tunneling on EX9200 switches, see "Configuring VLAN Encapsulation" on page 203 and related topics, or for other EX Series switches, see "Understanding Q-in-Q Tunneling and VLAN Translation" on page 554.

For non-ELS EX Series switches, L2PT is configured using statements in the **[edit vlans *vlan-name* dot1q-tunneling]** hierarchy, which means Q-in-Q tunneling is (and must be) enabled. If L2PT is not enabled, Layer 2 PDUs are handled in the same way they were handled before L2PT was enabled. For details on configuring L2PT on non-ELS EX Series switches, see "Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)" on page 395

NOTE: If the switch receives untagged or priority-tagged Layer 2 control PDUs to be tunneled, then you must configure the switch to map untagged and priority-tagged packets to an L2PT-enabled VLAN. For more information on assigning untagged and priority-tagged packets to VLANs, see "Understanding Q-in-Q Tunneling and VLAN Translation" on page 554 and "Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure)" on page 582.

For ELS EX Series switches, L2PT is configured using statements in the **[edit layer2-control mac-rewrite interface *interface-name*]** hierarchy to enable MAC address rewriting for Layer 2 protocol tunneling for a configured Q-in-Q interface. For details, see "Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure)" on page 398.

Related Documentation

- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- Configuring VLAN Encapsulation on page 203

Clearing a MAC Rewrite Error on an Interface with Layer 2 Protocol Tunneling

On EX9200 switches with Layer 2 protocol tunneling (L2PT) configured, customer-facing ports should not receive packets with the L2PT MAC address as the destination address unless there is a network topology or configuration error. Under these conditions, when an interface with L2PT enabled receives an L2PT packet, the interface state is disabled due to a MAC rewrite error, and must subsequently be re-enabled to continue operation.

1. To check whether an interface with L2PT enabled has become disabled due to a MAC rewrite error condition, use the **show interfaces** operational command:

   user@switch> **show interfaces *interface-name***

If the interface status includes **Disabled, Physical link is Down** and the **MAC-REWRITE Error** field is **Detected**, then a MAC rewrite error was detected and contributed to the interface being down. When no MAC rewrite error was detected, the **MAC-REWRITE Error** field is **None**.

For example, the following output shows a MAC rewrite error was detected on the given interface:

```
user@switch> show interfaces ge-0/0/2
Physical interface: ge-0/0/2, Disabled, Physical link is Down
  Interface index: 150, SNMP ifIndex: 531
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 1000mbps, BPDU
Error: None,
  Loop Detect PDU Error: None, Ethernet-Switching Error: None, Source filtering:
 Disabled
  Ethernet-Switching Error: None, MAC-REWRITE Error: Detected, Loopback:
Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, Media
 type: Fiber
  Device flags   : Present Running
```

2. To clear a MAC rewrite error from an interface that has L2PT enabled, use the **clear error mac-rewrite** operational command:

   user@switch> clear error mac-rewrite *interface-name*

Related Documentation

- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398
- show mac-rewrite interface on page 1395
- clear error mac-rewrite on page 1169

Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style.

Layer 2 protocol tunneling (L2PT) allows you to send Layer 2 protocol data units (PDUs) across a service provider network and deliver them to EX Series switches at a remote location. This feature is useful when you have a network that includes remote sites that are connected across a service provider network and you want to run Layer 2 protocols on switches connected across the service provider network.

Tunneled Layer 2 PDUs do not normally arrive at high rate. If the tunneled Layer 2 PDUs do arrive at high rate, there might be a problem in the network. Typically, you would want to shut down the interface that is receiving a high rate of tunneled Layer 2 PDUs so that the problem can be isolated. You do so using the **shutdown-threshold** statement. However, if you do not want to completely shut down the interface, you can configure the switch to drop tunneled Layer 2 PDUs that exceed a certain threshold using the **drop-threshold** statement.

There are no default settings for **drop-threshold** and **shutdown-threshold**. If you do not specify these thresholds, then no thresholds are enforced. As a result, the switch tunnels all Layer 2 PDUs regardless of the speed at which they are received, although the number of packets tunneled per second might be limited by other factors.

You can specify a drop threshold value without specifying a shutdown threshold value, and you can specify a shutdown threshold value without specifying a drop threshold value. If you specify both threshold values, then the drop threshold value must be less than or equal to the shutdown threshold value. If the drop threshold value is greater than the shutdown threshold value and you try to commit the configuration, the commit will fail.

NOTE: L2PT and VLAN translation configured with the mapping statement cannot both be configured on the same switch.

NOTE: If the switch receives untagged Layer 2 control PDUs to be tunnelled, then you must configure the switch to map untagged (native) packets to an L2PT-enabled VLAN. Otherwise, the untagged Layer 2 control PDU packets are discarded. For more information, see "Understanding Q-in-Q Tunneling and VLAN Translation" on page 554 and "Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure)" on page 582.

To configure L2PT on an EX Series switch:

1. Because L2PT operates under the Q-in-Q tunneling configuration, you must enable Q-in-Q tunneling before you can configure L2PT. Enable Q-in-Q tunneling on VLAN **customer-1**:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling
   ```

2. Enable L2PT for the Layer 2 protocol you want to tunnel, on the VLAN:

   - To enable L2PT for a specific protocol (here, STP):

     ```
     [edit]
     user@switch# set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp
     ```

   - To enable L2PT for all supported protocols:

     ```
     [edit]
     user@switch# set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling all
     ```

3. (Optional) Configure the drop threshold:

NOTE: If you also configure the shutdown threshold, ensure that you configure the drop threshold value to be less than or equal to the shutdown threshold value. If the drop threshold value is greater than the shutdown threshold value and you to try to commit the configuration changes, the commit will fail.

```
[edit]
user@switch# set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp
drop-threshold 50
```

4. (Optional) Configure the shutdown threshold:

NOTE: If you also configure the drop threshold, ensure that you configure the shutdown threshold value to be greater than or equal to the drop threshold value. If the shutdown threshold value is less than the drop threshold value and you to try to commit the configuration changes, the commit will fail.

```
[edit]
user@switch# set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp
shutdown-threshold 100
```

NOTE: Once an interface is disabled, you must explicitly reenable it using the clear ethernet-switching layer2-protocol-tunneling error command. Otherwise, the interface remains disabled.

Related Documentation

- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389

Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure)

NOTE: This topic applies to Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure)" on page 395. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

Layer 2 protocol tunneling (L2PT) enables you to send Layer 2 protocol data units (PDUs) across a service provider network and deliver them to EX Series switches at a remote location. This feature is useful when you have a network that includes remote sites that are connected across a service provider network and you want to run Layer 2 protocols on switches connected across the service provider network.

You can also use L2PT to tunnel protocols between two locally-connected user-to-network interfaces (UNIs) in the same broadcast domain, but in that case, the protocol packets are simply flooded in the VLAN instead of being rewritten with the tunnel MAC address.

To configure L2PT on an EX Series switch, you must first configure a Q-in-Q interface or group of interfaces. For information about configuring Q-in-Q tunneling on EX9200 switches, see "Configuring VLAN Encapsulation" on page 203, "Configuring Inner and Outer TPIDs and VLAN IDs" on page 204, and "Stacking a VLAN Tag" on page 208. For information about configuring Q-in-Q tunneling on other EX Series switches, see "Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure)" on page 583.

NOTE: When you enable L2PT tunneling for a protocol on one user-to-network interface (UNI) in a bridge domain or VLAN, all UNIs in the bridge domain or VLAN should also be configured to tunnel the same protocol for consistent behavior. In that case, those UNIs can receive non-tunneled packets, and tunneled packets are forwarded through the network-to-network interfaces (NNIs).

- To configure L2PT on a specified Q-in-Q interface, enable MAC address rewriting for Layer 2 protocol tunneling and select the Layer 2 protocol to be tunneled from the list of available options for the type of switch being configured (see **protocol**):

  ```
  [edit protocols]
  user@switch# set layer2-control mac-rewrite interface interface-name protocol protocol-name
  ```

  NOTE: You can select only one Layer 2 protocol at a time. If you want an interface to support tunneling more than one Layer 2 protocol, you must enter the mac-rewrite statement multiple times to select the desired protocols.

For example, on an EX9200 switch, the following commands configure a UNI (**xe-1/1/3**) for Q-in-Q tunneling and MAC address rewriting for STP:

```
set interfaces xe-1/1/3 flexible-vlan-tagging
set interfaces xe-1/1/3 encapsulation extended-vlan-bridge
set interfaces xe-1/1/3 unit 0 encapsulation vlan-bridge
set interfaces xe-1/1/3 unit 0 vlan-id 10
set interfaces xe-1/1/3 native-vlan-id 10
set interfaces xe-1/1/3 unit 0 input-vlan-map push
set interfaces xe-1/1/3 unit 0 input-vlan-map vlan-id 100
set interfaces xe-1/1/3 unit 0 output-vlan-map pop
set protocols layer2-control mac-rewrite interface xe-1/1/3 protocol stp
set vlans v10 interface xe-1/1/3.10
```

On an EX2300, EX3400, EX4300, or EX4600 switch, the following commands configure a UNI (**ge-0/0/0**) for Q-in-Q tunneling and MAC address rewriting for STP and LLDP:

```
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 encapsulation extended-vlan-bridge
set interfaces ge-0/0/0 unit 10 vlan-id 10
set interfaces ge-0/0/0 native-vlan-id 10
set interfaces ge-0/0/0 unit 10 input-vlan-map push
set interfaces ge-0/0/0 unit 10 output-vlan-map pop
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol stp
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol lldp
set vlans v10 interface ge-0/0/0.10
```

In the case where you want to tunnel protocols to or from two locally-connected UNIs on the same switch, although you still configure the **mac-rewrite** statement to specify the protocol being tunneled, the switch simply floods the protocol packets within the VLAN instead of rewriting the MAC address. The same configuration is used for both interfaces, and you do not need to use a loopback cable. For example, the following commands configure two UNIs (**ge-0/0/0** and **ge-0/0/1**) for Q-in-Q tunneling on an EX2300, EX3400, EX4300, or EX4600 switch, and LACP and LLDP packets are exchanged between the two ports on the switch:

```
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 native-vlan-id 20
set interfaces ge-0/0/0 encapsulation extended-vlan-bridge
set interfaces ge-0/0/0 unit 20 vlan-id 20
set interfaces ge-0/0/0 unit 20 input-vlan-map push
set interfaces ge-0/0/0 unit 20 output-vlan-map pop
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 native-vlan-id 20
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 20 vlan-id 20
set interfaces ge-0/0/1 unit 20 input-vlan-map push
set interfaces ge-0/0/1 unit 20 output-vlan-map pop
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol lacp
set protocols layer2-control mac-rewrite interface ge-0/0/0 protocol lldp
set protocols layer2-control mac-rewrite interface ge-0/0/1 protocol lacp
set protocols layer2-control mac-rewrite interface ge-0/0/1 protocol lldp
set vlans v10 interface ge-0/0/0.20
set vlans v10 interface ge-0/0/1.20
```

- To check the protocols that L2PT is configured to tunnel on an interface, enter the **show mac-rewrite interface** command in operational mode and specify the interface name, as follows:

```
user@switch> show mac-rewrite interface ge-0/0/0
Interface       Protocols
 ge-0/0/0        LLDP STP
```

  If you do not specify an interface name, the **show mac-rewrite interface** command displays all interfaces with L2PT configured. For example:

```
user@switch> show mac-rewrite interface
Interface       Protocols
 ge-0/0/0        LACP LLDP
 ge-0/0/1        LACP LLDP
```

- (EX9200 switches only) For information on how to detect and clear an interface configured for L2PT that appears to be blocked due to a MAC rewrite error, see "Clearing a MAC Rewrite Error on an Interface with Layer 2 Protocol Tunneling" on page 394.

Related Documentation

- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- Configuring VLAN Encapsulation on page 203
- Stacking a VLAN Tag on page 208

Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches

NOTE: This task uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style.

Layer 2 protocol tunneling (L2PT) allows service providers to send Layer 2 protocol data units (PDUs) across the provider's cloud and deliver them to EX Series switches that are not part of the local broadcast domain. This feature is useful when you want to run Layer 2 protocols on a network that includes switches located at remote sites that are connected across a service provider network.

NOTE: L2PT and VLAN translation configured with the mapping statement cannot both be configured on the same VLAN. However, L2PT can be configured on one VLAN on a switch while VLAN translation can be configured on a different VLAN that has no L2PT.

This example describes how to configure L2PT:

Requirements

This example uses the following hardware and software components:

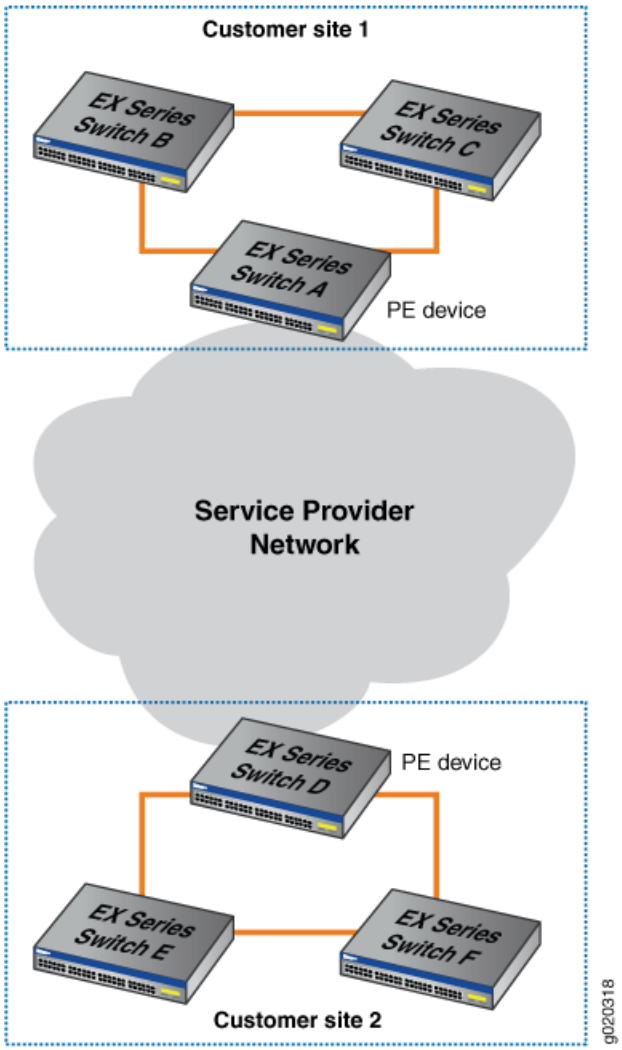
- Six EX Series switches, with three each at two customer sites, with one of the switches at each site designated as the provider edge (PE) device
- Junos OS Release 10.0 or later for EX Series switches

Overview and Topology

L2PT allows you to send Layer 2 PDUs across a service provider network and deliver them to EX Series switches that are not part of the local broadcast domain.

Figure 23 on page 392 shows a customer network that includes two sites that are connected across a service provider network. Site 1 contains three switches connected in a Layer 2 network, with Switch A designated as a provider edge (PE) device in the service provider network. Site 2 contains a Layer 2 network with a similar topology to that of Site 1, with Switch D designated as a PE device.

Figure 24: L2PT Topology



When you enable L2PT on a VLAN, Q-in-Q tunneling is also (and must be) enabled. Q-in-Q tunneling ensures that Switches A, B, C, D, E, and F are part of the same broadcast domain.

This example uses STP as the Layer 2 protocol being tunneled, but you could substitute any of the supported protocols for STP. You can also use the **all** keyword to enable L2PT for all supported Layer 2 protocols.

Tunneled Layer 2 PDUs do not normally arrive at a high rate. If the tunneled Layer 2 PDUs do arrive at a high rate, there might be a problem in the network. Typically, you would want to shut down the interface that is receiving a high rate of tunneled Layer 2 PDUs so that the problem can be isolated. Alternately, if you do not want to completely shut down the interface, you can configure the switch to drop tunneled Layer 2 PDUs that exceed a certain threshold.

The **drop-theshold** configuration statement allows you to 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 must be less than or equal to the shutdown threshold. If the drop threshold is greater than the shutdown threshold and you try to commit the configuration, the commit will fail.

The **shutdown-threshold** configuration statement allows you to 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 specified interface is disabled. The shutdown threshold must be greater than or equal to the drop threshold. You can specify a drop threshold without specifying a shutdown threshold, and you can specify a shutdown threshold without specifying a drop threshold. If you do not specify these thresholds, then no thresholds are enforced. As a result, the switch tunnels all Layer 2 PDUs regardless of the speed at which they are received, although the number of packets tunneled per second might be limited by other factors.

In this example, we will configure both a drop threshold and a shutdown threshold to show how this is done.

If L2PT-encapsulated packets are received on an access interface, the switch reacts as it does when there is a loop between the service provider network and the customer network and shuts down (disables) the access interface.

Once an interface is disabled, you must explicitly reenable it using the **clear ethernet-switching layer2-protocol-tunneling error** command or else the interface will remain disabled.

Configuration

To configure L2PT, perform these tasks:

CLI Quick Configuration

To quickly configure L2PT, copy the following commands and paste them into the switch terminal window of each PE device (in Figure 23 on page 392, Switch A and Switch D are the PE devices):

```
[edit]
set vlans customer-1 dot1q-tunneling
set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp
set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp drop-threshold 50
set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp shutdown-threshold 100
```

Step-by-Step Procedure

To configure L2PT, perform these tasks on each PE device (in Figure 23 on page 392, Switch A and Switch D are the PE devices):

1. Enable Q-in-Q tunneling on VLAN **customer-1**:

   ```
   [edit]
   user@switch# set vlans customer-1 dot1q-tunneling
   ```

2. Enable L2PT for STP on VLAN **customer-1**:

```
[edit]
user@switch#  set vlans customer-1 dot1q-tunneling layer2-protocol-tunneling stp
```

3. Configure the drop threshold as **50**:

```
[edit]
user@switch#  set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp
drop-threshold 50
```

4. Configure the shutdown threshold as **100**:

```
[edit]
user@switch#  set vlans customer-1 dot1q-tunneling layer2–protocol-tunneling stp
shutdown-threshold 100
```

Results Check the results of the configuration:

```
[edit]
user@switch# show vlans customer-1 dot1q-tunneling
layer2-protocol-tunneling {
   stp {
      drop-threshold 50;
      shutdown-threshold 100;
   }
}
```

Verification

To verify that L2PT is working correctly, perform this task:

- Verify That L2PT Is Working Correctly on page 404

Verify That L2PT Is Working Correctly

Purpose Verify that Q-in-Q tunneling and L2PT are enabled.

Action Check to see that Q-in-Q tunneling and L2PT are enabled on each PE device (Switch A and Switch D are the PE devices):

```
user@switchA> show vlans extensive customer-1
VLAN: customer-1, Created at: Thu Jun 25 05:07:38 2009
802.1Q Tag: 100, Internal index: 4, Admin State: Enabled, Origin: Static
Dot1q Tunneling status: Enabled
Layer2 Protocol Tunneling status: Enabled
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  3 (Active = 0)
      ge-0/0/7.0, untagged, access
      ge-0/0/8.0, untagged, access
      ge-0/0/9.0, untagged, access
```

Check to see that L2PT is tunneling STP on VLAN **customer-1** and that **drop-threshold** and **shutdown-threshold** have been configured:

```
user@switchA> show ethernet-switching layer2-protocol-tunneling vlan customer-1

Layer2 Protocol Tunneling VLAN information:
VLAN          Protocol      Drop       Shutdown
                            Threshold  Threshold
customer-1    stp           50         100
```

Check the state of the interfaces on which L2PT has been enabled, including what kind of operation (encapsulation or decapsulation) they are performing:

```
user@switchA> show ethernet-switching layer2-protocol-tunneling interface

Layer2 Protocol Tunneling information:
Interface     Operation       State      Description
ge-0/0/0.0    Encapsulation   Shutdown   Shutdown threshold exceeded
ge-0/0/1.0    Decapsulation   Shutdown   Loop detected
ge-0/0/2.0    Decapsulation   Active
```

Meaning The **show vlans extensive customer-1** command shows that Q-in-Q tunneling and L2PT have been enabled. The **show ethernet-switching layer2-protocol-tunneling vlan customer-1** command shows that L2PT is tunneling STP on VLAN **customer-1**,the drop threshold is set to **50**, and the shutdown threshold is set to **100**. The **show ethernet-switching layer2-protocol-tunneling interface** command shows the type of operation being performed on each interface, the state of each interface and, if the state is **Shutdown**, the reason why the interface is shut down.

Related Documentation

- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389

CHAPTER 17

Configuring Ethernet Ring Protection

Ethernet Ring Protection Switching Overview

Ethernet ring protection switching (ERPS) helps achieve high reliability and network stability. Links in the ring will never form loops that fatally affect the network operation and services availability. The basic idea of an Ethernet ring is to use one specific link to protect the whole ring. This special link is called a *ring protection link (RPL)*. If no failure happens in other links of the ring, the RPL blocks the traffic and is not used. The RPL is controlled by a special node called an *RPL owner*. There is only one RPL owner in a ring. The RPL owner is responsible for blocking traffic over the RPL. Under ring failure conditions, the RPL owner is responsible for unblocking traffic over the RPL. A ring failure results in protection switching of the RPL traffic. An automatic protection switching (APS) protocol is used to coordinate the protection actions over the ring. Protection switching blocks traffic on the failed link and unblocks the traffic on the RPL. When the failure clears, revertive protection switching blocks traffic over the RPL and unblocks traffic on the link on which the failure is cleared.

NOTE: **ERPS on AE interfaces is not supported on ACX Series routers except on ACX5000 Series routers.**

The following standards provide detailed information on Ethernet ring protection switching:

- ITU-T Recommendation G.8032/Y.1344 version 1 and 2, *Ethernet Ring protection switching*. G.8032v1 supports a single ring topology and G.8032v2 supports multiple rings and ladder topology.

NOTE: EX2300 and EX3400 switches support G.8032v1 only.

- ITU-T Y.1731, *OAM functions and mechanisms for Ethernet-based networks*

For additional information on configuring Ethernet ring protection switching on EX Series switches, see "Example: Configuring Ethernet Ring Protection Switching on EX Series Switches" on page 420.

For additional information on configuring Ethernet ring protection switching on MX Series routers, see the *Layer 2 Configuration Guide* for a complete example of Ethernet rings and information about STP loop avoidance and prevention.

Related Documentation

- Understanding Ethernet Ring Protection Switching Functionality on page 408
- *Configuring Ethernet Ring Protection Switching*
- *Example: Ethernet Ring Protection Switching Configuration on MX Routers*
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- *Ethernet Interfaces Feature Guide for Routing Devices*

Understanding Ethernet Ring Protection Switching Functionality

Acronyms

The following acronyms are used in the discussion about Ethernet ring protection switching (ERPS):

- MA—Maintenance association
- MEP—Maintenance association end point
- OAM—Operations, administration, and management (Ethernet ring protection switching uses connectivity fault management daemon)
- FDB—MAC forwarding database
- STP—Spanning Tree Protocol
- RAPS—Ring automatic protection switching
- WTB—Wait to block. Note that WTB is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting on EX2300 and EX3400 switches has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect on EX2300 and EX4300 switches.
- WTR—Wait to restore. Note that on EX2300 and EX3400 switches only, the WTR configuration must be 5-12 minutes.
- RPL—Ring protection link

Ring Nodes

Multiple nodes are used to form a ring. There are two different node types:

- Normal node—The node has no special role on the ring.
- RPL owner node—The node owns the RPL and blocks or unblocks traffic over the RPL.

Ring Node States

The following are the different states for each node of a specific ring:

- init—Not a participant of a specific ring.
- idle—No failure on the ring; the node is performing normally. For a normal node, traffic is unblocked on both ring ports. For the RPL owner or RPL neighbor, traffic is blocked on the ring port that connects to the RPL and unblocked on the other ring port.
- protection—A failure occurred on the ring. For a normal node, traffic is blocked on the ring port that connects to the failing link and unblocked on working ring ports. For the RPL owner, traffic is unblocked on both ring ports if they connect to non-failure links.
- pending—The node is recovering from failure or its state after a **clear** command is used to remove the previous manual command. When a protection group is configured, the node enters the pending state. When a node is in pending state, the WTR or WTB timer will be running. All nodes are in pending state till WTR or WTB timer expiry.

- force switch—A force switch is issued. When a force switch is issued on a node in the ring all nodes in the ring will move into the force switch state.

  NOTE: EX2300 and EX3400 switches do not support force switch.

- manual switch—A manual switch is issued. When a manual switch is issued on a node in the ring all nodes in the ring will move into the manual switch state.

  NOTE: EX2300 and EX3400 switches do not support manual switch.

There can be only one RPL owner for each ring. The user configuration must guarantee this, because the APS protocol cannot check this.

Default Logging of Basic State Transitions on EX Series Switches

Starting with Junos OS Release 14.1X53-D15, EX Series switches automatically log basic state transitions for the ERPS protocol. Starting with Junos OS Release 18.2R1, EX2300 and EX3400 switches automatically log basic state transitions for the ERPS protocol. No configuration is required to initiate this logging. Basic state transitions include ERPS interface transitions from up to down, and down to up; and ERPS state transitions from idle to protection, and protection to idle.

The basic state transitions are logged in a single file named **erp-default**, which resides in the **/var/log** directory of the switch. The maximum size of this file is 15 MB.

Default logging for ERPS can capture initial ERPS interface and state transitions, which can help you troubleshoot issues that occur early in the ERPS protocol startup process. However, if more robust logging is needed, you can enable traceoptions for ERPS by entering the **traceoptions** statement in the **[edit protocols protection-group]** hierarchy.

Be aware that for ERPS, only default logging or traceoptions can be active at a time on the switch. That is, default logging for ERPS is automatically enabled and if you enable traceoptions for ERPS, the switch automatically disables default logging. Conversely, if you disable traceoptions for ERPS, the switch automatically enables default logging.

Logical Ring

You can define multiple logical-ring instances on the same physical ring. The logical ring feature currently supports only the physical ring, which means that two adjacent nodes of a ring must be physically connected and the ring must operate on the physical interface, not the VLAN. Multiple ring instances are usually defined with trunk mode ring interfaces.

FDB Flush

When ring protection switching occurs, normally an *FDB flush* is executed. The Ethernet ring control module uses the same mechanism as the STP to trigger the FDB flush. The Ethernet ring control module controls the ring port physical interface's default STP index to execute the FDB flush.

NOTE: Optimized flushing is not supported on EX2300 and EX3400 switches.

Starting with Junos OS Release 14.2, the FDB flush depends on the RAPS messages received on the both the ports of the ring node.

Traffic Blocking and Forwarding

Ethernet ring control uses the same mechanism as the STP to control forwarding or discarding of user traffic. The Ethernet ring control module sets the ring port physical interface default STP index state to forwarding or discarding in order to control user traffic.

RPL Neighbor Node

Starting with Junos OS Release 14.2, ring protection link neighbor nodes are supported. An RPL neighbor node is adjacent to the RPL and is not the RPL owner. If a node is configured with one interface as the protection-link-end and no protection-link-owner is present in its configuration, the node is an RPL neighbor node.

NOTE: RPL neighbor node is not supported on EX2300 and EX3400 switches.

RAPS Message Blocking and Forwarding

The router or switch treats the ring automatic protection switching (RAPS) message the same as it treats user traffic for forwarding RAPS messages between two ring ports. The ring port physical interface default STP index state also controls forwarding RAPS messages between the two ring ports. Other than forwarding RAPS messages between the two ring ports, as shown in Figure 25 on page 411, the system also needs to forward the RAPS message between the CPU (Ethernet ring control module) and the ring port. This type of forwarding does not depend on the ring port physical interfaces' STP index state. The RAPS message is always sent by the router or switch through the ring ports, as shown in Figure 26 on page 411. A RAPS message received from a discarding ring port is sent to the Ethernet ring control module, but is not sent to the other ring port.

Figure 25: Protocol Packets from the Network to the Router

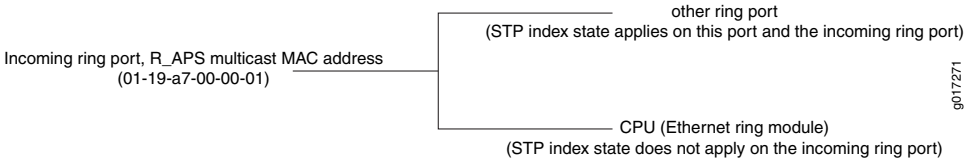
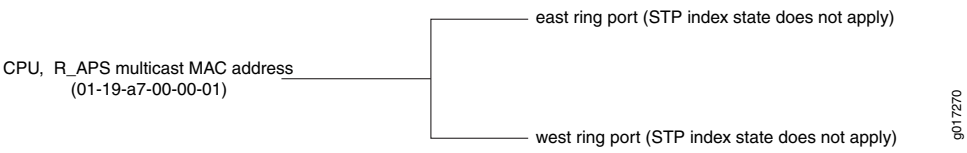


Figure 26: Protocol Packets from the Router or Switch to the Network



Juniper Networks switches and Juniper Networks routers use different methods to achieve these routes.

The switches use forwarding database entries to direct the RAPS messages. The forwarding database entry (keyed by the RAPS multicast address and VLAN) has a composite next hop associated with it—the composite next hop associates the two ring interfaces with the forwarding database entry and uses the split horizon feature to prevent sending the packet out on the interface that it is received on. This is an example of the forwarding database entry relating to the RAPS multicast MAC (a result of the **show ethernet-switching table detail** command):

```
VLAN: v1, Tag: 101, MAC: 01:19:a7:00:00:01, Interface: ERP
Interfaces:                  ge-0/0/9.0, ge-0/0/3.0
Type: Static
Action: Mirror
Nexthop index: 1333
```

The routers use an implicit filter to achieve ERP routes. Each implicit filter binds to a bridge domain. Therefore, the east ring port control channel and the west ring port control channel of a particular ring instance must be configured to the same bridge domain. For each ring port control channel, a filter term is generated to control RAPS message forwarding. The filter number is the same as the number of bridge domains that contain the ring control channels. If a bridge domain contains control channels from multiple rings, the filter related to this bridge domain will have multiple terms and each term will relate to a control channel. The filter has command parts and control-channel related parts, as follows:

- Common terms:
  - ```
    term 1: if [Ethernet type is not OAM Ethernet type (0x8902)
    ]              { accept packet }
    ```
  - ```
    term 2: if [source MAC address belongs to this bridge]
                { drop packet, our packet loop through the ring and come back
                    to home}
    ```
  - ```
    term 3: if [destination is the RAPS PDU multicast address(0x01,0x19,0xa7,
                    0x00,0x00,0x01] AND[ring port STP status is DISCARDING]
                 { send to CPU }
    ```
- Control channel related terms:
  - ```
          if [destination is the RAPS PDU multicast
    address(0x01,0x19,0xa7,0x00,0x00,
             0x01] AND[ring port STP status is FORWARDING] AND [Incoming
    interface
              IFL equal to control channel IFL]
             { send packet to CPU and send to the other ring port }
          default term: accept packet.
    ```

Dedicated Signaling Control Channel

For each ring port, a dedicated signaling control channel with a dedicated VLAN ID must be configured. In Ethernet ring configuration, only this control logical interface is configured and the underlying physical interface is the physical ring port. Each ring requires that two control physical interfaces be configured. These two logical interfaces must be configured

in a bridge domain for routers (or the same VLAN for switches) in order to forward RAPS protocol data units (PDUs) between the two ring control physical interfaces. If the router control channel logical interface is not a trunk port, only control logical interfaces will be configured in ring port configuration. If this router control channel logical interface is a trunk port, in addition to the control channel logical interfaces, a dedicated VLAN ID must be configured for routers. For switches, always specify either a VLAN name or VLAN ID for all links.

RAPS Message Termination

The RAPS message starts from the originating node, travels through the entire ring, and terminates in the originating node unless a failure is present in the ring. The originating node must drop the RAPS message if the source MAC address in the RAPS message belongs to itself. The source MAC address is the node's node ID.

Revertive and Non-revertive Modes

In revertive operation, once the condition causing a switch has cleared, traffic is blocked on the RPL and restored to the working transport entity. In nonrevertive operation, traffic is allowed to use the RPL if it has not failed, even after a switch condition has cleared.

NOTE: **Non-revertive mode is not supported on EX2300 and EX3400 switches.**

Multiple Rings

The Ethernet ring control module supports multiple rings in each node (two logical interfaces are part of each ring). The ring control module also supports the interconnection of multiple rings. Interconnection of two rings means that two rings might share the same link or share the same node. Ring interconnection is supported only using non-virtual-channel mode. Ring interconnection using virtual channel mode is not supported.

NOTE: **Interconnection of multiple rings is not supported on EX2300 and EX3400 switches.**

Node ID

For each node in the ring, a unique *node ID* identifies each node. The node ID is the node's MAC address.

For routers only, you can configure this node ID when configuring the ring on the node or automatically select an ID like STP does. In most cases, you will not configure this and the router will select a node ID, like STP does. It should be the manufacturing MAC address. The ring node ID should not be changed, even if you change the manufacturing MAC address. Any MAC address can be used if you make sure each node in the ring has a different node ID. The node ID on switches is selected automatically and is not configurable.

Ring ID

The ring ID is used to determine the value of the last octet of the MAC destination address field of the RAPS protocol data units (PDUs) generated by the ERP control process. The ring ID is also used to discard any RAPS PDU, received by this ERP control process with a non-matching ring ID. Ring ID values 1 through 239 are supported.

Bridge Domains with the Ring Port (MX Series Routers Only)

On the routers, the protection group is seen as an abstract logical port that can be configured to any bridge domain. Therefore, if you configure one ring port or its logical interface in a bridge domain, you must configure the other related ring port or its logical interface to the same bridge domain. The bridge domain that includes the ring port acts as any other bridge domain and supports the IRB Layer 3 interface.

Wait-to-Block Timer

The RPL owner node uses a delay timer before initiating an RPL block in revertive mode of operation or before reverting to IDLE state after clearing manual commands. The Wait-to-Block (WTB) timer is used when clearing **force switch** and **manual switch** commands. As multiple **force switch** commands are allowed to coexist in an Ethernet ring, the WTB timer ensures that clearing of a single **force switch** command does not trigger the re-blocking of the RPL. When clearing a **manual switch** command, the WTB timer prevents the formation of a closed loop due to a possible timing anomaly where the RPL Owner Node receives an outdated remote **manual switch** request during the recovery process.

When recovering from a **manual switch** command, the delay timer must be long enough to receive any latent remote **force switch**, signal failure, or **manual switch** commands. This delay timer is called the WTB timer and is defined to be 5 seconds longer than the guard timer. This delay timer is activated on the RPL Owner Node. When the WTB timer expires, the RPL Owner Node initiates the reversion process by transmitting an RAPS (NR, RB) message. The WTB timer is deactivated when any higher-priority request preempts it.

NOTE: The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.

Adding and Removing a Node

Starting with Junos OS Release 14.2, you can add or remove a node between two nodes in an Ethernet ring. Nodes are added or removed using the **force switch** command.

NOTE: EX2300 and EX3400 switches do not support force switch.

Release History Table

Release	Description
18.2R1	Starting with Junos OS Release 18.2R1, EX2300 and EX3400 switches automatically log basic state transitions for the ERPS protocol.
14.2	Starting with Junos OS Release 14.2, the FDB flush depends on the RAPS messages received on the both the ports of the ring node.
14.2	Starting with Junos OS Release 14.2, ring protection link neighbor nodes are supported.
14.2	Starting with Junos OS Release 14.2, you can add or remove a node between two nodes in an Ethernet ring.
14.1X53-D15	Starting with Junos OS Release 14.1X53-D15, EX Series switches automatically log basic state transitions for the ERPS protocol.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- *Configuring Ethernet Ring Protection Switching*
- *Example: Ethernet Ring Protection Switching Configuration on MX Routers*
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure)

You can configure Ethernet ring protection switching (ERPS) on connected switches to prevent fatal loops from disrupting a network. ERPS is similar to spanning-tree protocols, but ERPS is more efficient than spanning-tree protocols because it is customized for ring topologies. You must configure at least three switches to form a ring. One of the links, called the ring protection link (RPL) end interface, is blocked until another link fails—at this time the RPL link is unblocked, ensuring connectivity.

NOTE: In legacy EX Series switches, ERPS acts when the logical interface goes up or down. On EX4300 and QFX Series switches, ERPS acts only when the physical interface goes up or down. Therefore, ERPS does not react to the connectivity fault management (CFM) logical interface when it goes up or down.

NOTE: Ethernet OAM connectivity fault management (CFM) can be used with ERPS to detect link faults faster in some cases. See *Configuring Ethernet OAM Connectivity Fault Management (CLI Procedure)*.

The time needed for switchover to the ERPS link is affected by three settings—link failure detection time, the number of nodes in the ring, and the time it takes to unblock the RPL after a failure is detected.

NOTE: Do not configure redundant trunk groups on ERPS interfaces. You can configure VSTP on ERPS interfaces if the VSTP uses a VLAN that is not part of the ERPS control VLAN or data channel VLANs. The total number of ERPS and VSTP or MSTP instances is limited to 253.

NOTE:

- On EX2300 and EX3400 switches, if ERP is configured on an interface, that interface cannot participate in STP, RSTP or MSTP. If a VLAN is part of an MST instance, it cannot also be in ERP's control VLAN or data VLAN. Also, due to TCAM size, the combined total number of ERP instances and MSTP instances, or ERP instances alone, is limited to 50 instances for the EX2300 switch and 100 instances for the EX3400 switch.
-

Before you begin:

- Configure a VLAN to act as a control channel for ERPS. Two interfaces (east and west) on each switch in the ring must be associated with the control VLAN. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98.

NOTE: When EX2300 and EX3400 ERPS switches have a VLAN-ID configured with a name under an interface hierarchy, a commit error occurs. Avoid this by configuring VLAN-IDs using numbers when they are under an interface hierarchy with ERPS configured in the switch.

  - The interfaces on the ERPS control channel are usually (but not required to be) configured as trunk ports. See *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*. Note that if one switch has trunk ports as the ERPS control interfaces, the same must be true of all switches on the ring (the ERPS control interfaces must also be trunk ports).

- Data channels are optional on the ERPS link. If you plan to use them, configure a VLAN for each data channel. If you have multiple ERPS instances, the control VLANs and data channel VLANs must not overlap.

To configure ERPS:

NOTE: You must configure at least three switches, with only one switch designated as the RPL owner node.

1. Spanning tree protocols and ERPS cannot both be configured on the ring ports, so on each ERPS interface, you must disable any configured spanning tree protocols (such as STP, RSTP, VSTP, or MSTP). Spanning tree protocols are disabled for individual interfaces in two different ways, depending on which Junos OS version and release is running on the switch. RSTP is enabled in the default configuration, so disabling RSTP is shown here.

   For switches without Enhanced Layer 2 (ELS) software support, and switches running ELS software with Junos OS release 15.1 or later, use this command to disable RSTP on the individual ERPS interfaces:

   ```
   [edit protocols]
   user@switch# set rstp interface interface-name disable
   ```

   For switches running Enhanced Layer 2 (ELS) software with Junos OS releases prior to 15.1, you disable spanning tree protocols on individual interfaces by deleting that configuration item. Use this command to delete the RSTP configuration item on the individual ERPS interfaces:

   ```
   [edit protocols]
   user@switch# delete rstp interface interface-name
   ```

2. Create a node ring on each switch:

   ```
   [edit protocols]
   user@switch# set protection-groupethernet-ring ring-name
   ```

3. Configure a control VLAN for the node ring:

   ```
   [edit protocols protection-group ethernet-ring ring name]
   ```

```
user@switch# set control-vlan vlan-name-or-vlan-id
```

4. Configure the east and west interfaces of the node ring with the control-channel interface.

```
[edit protocols protection-group ethernet-ring ring-name]
user@switch# set east-interface control-channel control-channel-name
user@switch# set west-interface control-channel control-channel-name
```

For switches with ELS support, additionally associate the east and west interfaces with the control VLAN:

```
[edit protocols protection-group ethernet-ring ring-name]
user@switch# set east-interface control-channel vlan vlan-name-or-vlan-id
user@switch# set west-interface control-channel vlan vlan-name-or-vlan-id
```

5. In addition, configure either the east interface or the west interface (but not both) as a link end. For example, configure the east interface:

```
[edit protocols protection-group ethernet-ring ring-name]
user@switch# set east-interface ring-protection-link-end
```

6. Configure only one switch as the RPL owner node:

```
[edit protocols protection-group ethernet-ring ring-name]
user@switch# set ring-protection-link-owner
```

7. The restore interval is the time the RPL owner node waits after the last ring automatic protection switching (RAPS) signal failure (SF) event has been cleared, to see if any further RAPS events occur. During this time interval, the RPL owner continues to process RAPS packets, and the ring remains in protection state with the RPL link unblocked. When this interval expires, if no further RAPS SF events have been reported, the RPL owner reverts the protection switching, blocks the RPL link, and returns the protection ring to idle state. Optionally, configure a local restore interval for the ERPS ring on each switch:

```
[edit protocols protection-group ethernet-ring ring-name]
user@switch# set restore-interval restore-interval-value
```

NOTE: The restore interval can also be set globally to apply to any ERPS rings configured on the switch. Local per-ring settings take priority over global settings.

8. The guard interval prevents ring nodes from receiving outdated RAPS messages. Optionally, configure the guard interval on each switch:

```
[edit protocols protection-group ethernet-ring ring name]
user@switch# set guard-interval guard-interval-value
```

NOTE: The guard interval can also be set globally to apply to any ERPS rings configured on the switch. Local per-ring settings take priority over global settings.

9. Global restore and guard interval settings are used when no local settings are configured. If these intervals are not configured globally or locally, the default values apply. Optionally configure global interval settings on the switch to apply to all rings that do not have a corresponding interval configured locally for the ring:

   - restore interval:

     ```
     [edit protocols protection-group]
     user@switch# set restore-interval restore-interval-value
     ```

   - guard interval:

     ```
     [edit protocols protection-group]
     user@switch# set guard-interval guard-interval-value
     ```

   NOTE: You can also configure other global settings, such as ERP traceoptions (file, page size, file size, flag name).

10. Optionally, configure VLANs for data channels on the ERPS link:

    ```
    [edit protocols protection-group ethernet-ring ring name]
    user@switch# set data-channel vlan-name
    ```

    NOTE: G.8032v1 supports a single ring topology and G.8032v2 supports multiple rings and ladder topology. Because Junos OS uses an ERPV2 state machine for ERPV1 support on both EX2300 and EX3400 switches, how ERPS works on those two switches deviates from the ERPV1 ITU standard in the following ways:

    - Wait to Restore (WTR) configuration values on EX2300 and EX3400 switches must be 5-12 minutes.
    - The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration change has no effect on the WTB setting, although the output from the **show protection-group ethernet-ring node-state detail** CLI command lists a WTB setting. However, that setting has no effect.
    - During initial state machine initialization on EX2300 and EX3400 switches, both ERPV1 ring ports move to a discarding state on the non-RPL node. During ERPV1 initial state machine initialization on EX2300 and EX3400 switches, the Automatic Protection Switching (APS) state moves to an idle state on the non-RPL switch.

Related Documentation

- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Ethernet Ring Protection Switching Overview on page 407
- Understanding Ethernet Ring Protection Switching Functionality on page 408

Example: Configuring Ethernet Ring Protection Switching on EX Series Switches

You can configure Ethernet ring protection switching (ERPS) on connected EX Series or QFX Series switches to prevent fatal loops from disrupting a network. (Platform support depends on the Junos OS release in your installation.) ERPS is similar to spanning-tree protocols, but ERPS is more efficient because it is customized for ring topologies. You must configure at least three switches to form a ring.

This example shows how to configure Ethernet ring protection switching on four switches that are connected to one another on a dedicated link in a ring topology.

NOTE: This task uses Junos OS for EX Series switches without support for the Enhanced Layer 2 Software (ELS) configuration style. However, an ERPS ring can include different types of switches, with or without ELS support. If you are configuring an ERPS ring that also includes QFX Series or EX Series switches running software that supports ELS, see "Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS" on page 435 for equivalent example configuration steps on those switches. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

- Requirements on page 420
- Overview and Topology on page 421
- Configuration on page 423
- Verification on page 434

Requirements

This example uses the following hardware and software components:

- Four connected EX Series switches that will function as nodes in the ring topology.

NOTE: Because Junos uses an ERPV2 state machine for ERPV1 support on both EX2300 and EX3400 switches, operation of ERPS on those two switches deviates from the ERPV1 ITU standard in the following ways:

- Wait to Restore (WTR) configuration values on EX2300 and EX3400 switches must be 5-12 minutes.
- The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.
- During initial state machine initialization on EX2300 and EX3400 switches, both ERPV1 ring ports move to a discarding state on the non-RPL node.
- During ERPV1 initial state machine initialization on EX2300 and EX3400 switches, the Automatic Protection Switching (APS) state moves to an idle state on the non-RPL switch

- Junos OS Release 12.1 or later without support for the Enhanced Layer 2 Software (ELS) configuration style.

Before you begin, be sure you have:

- Configured two trunk interfaces on each of the four switches. See Table 75 on page 422 for a list of the interface names used in this example.
- Configured the same VLAN (**erp-control-vlan-1**) with ID 100 on all four switches and associated two network interfaces from each of the four switches with the VLAN. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98. See Table 75 on page 422 for a list of the interface names used in this example.
- Configured two VLANs (**erp-data-1** and **erp-data-2**) with IDs 101 and 102, respectively, on all four switches and associated both the east and west interfaces on each switch with **erp-data-1** and **erp-data-2**. See Table 75 on page 422 for a list of the interface names used in this example.

NOTE: When EX2300 and EX3400 ERPS switches have a VLAN-ID configured with a name under an interface hierarchy, a commit error occurs. Avoid this by configuring VLAN-IDs using numbers when they are under an interface hierarchy with ERPS configured in the switch.

Overview and Topology

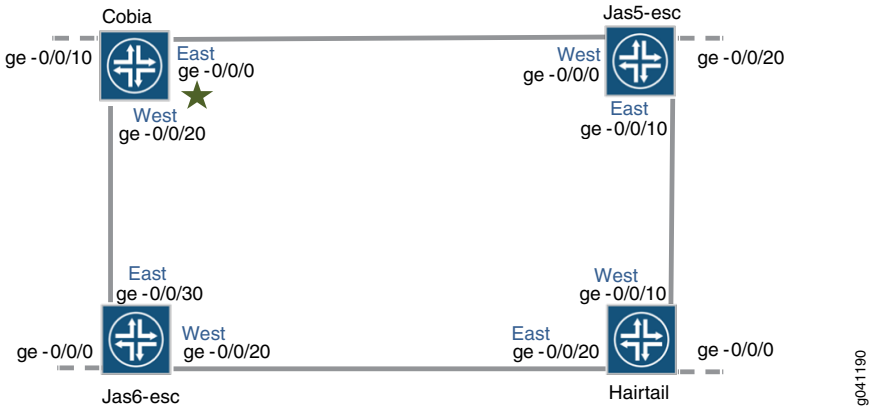
ERPS uses a dedicated physical link, including a control VLAN for trunk ports, between all of the switches to protect the active links. ERPS VLANs are all located on this link and are also blocked by default. When traffic between the switches is flowing with no

problems, the active links take care of all traffic. Only if an error occurs on one of the data links would the ERPS control channel take over and start forwarding traffic.

NOTE: Trunk ports on switches use a VLAN to create individual control channels for ERPS. When multiple ERPS instances are configured for a ring, there are multiple sets of ring protection links (RPLs) and RPL owners on the ERPS link, and a different channel is blocked for each instance. Nontrunk ports use the physical link as the control channel and protocol data units (PDUs) are untagged, with no VLAN information in the packet.

This example creates one protection ring (called a node ring) named erp1 on four switches connected in a ring by trunk ports as shown in Figure 27 on page 422. Because the links are trunk ports, the VLAN named erp-control-vlan-1 is used for erp1 traffic. The east interface of each switch is connected with the west interface of an adjacent switch. Cobia is the RPL owner, with interface ge-0/0/0 configured as an RPL end interface. The interface ge-0/0/0 of Jas5-esc is configured as the RPL neighbor interface. In the idle state, the RPL end blocks the control VLAN and data channel VLAN for this particular ERP instance—the blocked port on Cobia is marked with a star in Figure 27 on page 422.

Figure 27: Ethernet Ring Protection Switching Example



In this example, we configure the four switches with the interfaces indicated in both Figure 27 on page 422 and Table 75 on page 422.

Table 75: Components to Configure for This Example

Interfaces	Cobia	Jas5-esc	Jas6-esc	Hairtail
East	ge-0/0/0	ge-0/0/10	ge-0/0/30	ge-0/0/20
West	ge-0/0/20	ge-0/0/0	ge-0/0/20	ge-0/0/10
Third	ge-0/0/10	ge-0/0/20	ge-0/0/0	ge-0/0/0

Configuration

Configuring ERPS on Cobia, the RPL Owner Node

CLI Quick Configuration

To quickly configure Cobia, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

NOTE: Spanning-tree protocols and ERPS cannot both be configured on a ring port. Because RSTP is the spanning-tree protocol enabled in the default switch configuration, this example shows disabling RSTP on each ring port before configuring ERPS. If another spanning-tree protocol is enabled, you must disable that first instead.

```
set protocols rstp interface ge-0/0/0 disable
set protocols rstp interface ge-0/0/20 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 ring-protection-link-owner
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 east-interface ring-protection-link-end
set protocols protection-group ethernet-ring erp1 west-interface control-channel
  ge-0/0/20.0
```

Step-by-Step Procedure

To configure ERPS on Cobia:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. STP, RSTP, VSTP, and MSTP are all available spanning tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/0 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Designate Cobia as the RPL owner node:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set ring-protection-link-owner
```

4. Configure the VLANs erp-data-1 and erp-data-2 as data channels:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel erp-data-1
user@switch# set data-channel erp-data-2
```

5. Configure the control VLAN erp-control-vlan-1 for this ERP instance on the trunk interface:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan erp-control-vlan-1
```

6. Configure the east interface of the node ring erp1 with the control channel ge-0/0/0.0 and indicate that this particular ring protection link ends here:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/0.0
user@switch# set east-interface ring-protection-link-end
```

7. Configure the west interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/20.0
```

Results In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
  rstp {
      interface ge-0/0/20.0 {
          disable;
      }
      interface ge-0/0/0.0 {
          disable;
      }
  }
  protection-group {
      ethernet-ring erp1 {
          ring-protection-link-owner;
          east-interface {
              control-channel {
                  ge-0/0/0.0;
              }
              ring-protection-link-end;
          }
          west-interface {
              control-channel {
                  ge-0/0/20.0;
```

```
            }
        }
        control-vlan erp-control-vlan-1;
        data-channel {
            vlan [ 101-102 ];
        }
    }
}
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show vlans
 erp-control-vlan-1 {
    vlan-id 100;
    interface {
        ge-0/0/0.0;
        ge-0/0/20.0;
    }
 }
 erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/10.0;
        ge-0/0/0.0;
        ge-0/0/20.0;
    }
 }
 erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/10.0;
        ge-0/0/0.0;
        ge-0/0/20.0;
    }
 }
```

In configuration mode, check your interface configurations by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show interfaces
 ge-0/0/0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
 }
 ge-0/0/10 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
        }
    }
 }
 ge-0/0/20 {
```

```
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
```

If you are finished configuring the device, enter **commit** in configuration mode.

Configuring ERPS on Jas5-esc

CLI Quick Configuration

To quickly configure Jas5-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols rstp interface ge-0/0/10 disable
set protocols rstp interface ge-0/0/0 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/10.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel
  ge-0/0/0.0
```

Step-by-Step Procedure

To configure ERPS on Jas5-esc:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/10 disable
   user@switch# set rstp interface ge-0/0/0 disable
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure a control VLAN named erp-control-vlan-1 for the node ring erp1:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan erp-control-vlan-1
   ```

4. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance.

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel erp-data-1
   user@switch# set data-channel erp-data-2
   ```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/10.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/10.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/0.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/0.0
```

Results In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
  rstp {
      interface ge-0/0/10.0 {
          disable;
      }
      interface ge-0/0/0.0 {
          disable;
      }
  }
  protection-group {
      ethernet-ring erp1 {
          east-interface {
              control-channel {
                  ge-0/0/10.0;
              }
          }
          west-interface {
              control-channel {
                  ge-0/0/0.0;
              }
          }
          control-vlan erp-control-vlan-1;
          data-channel {
              vlan [ 101-102 ];
          }
      }
  }
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show vlans
  erp-control-vlan-1 {
      vlan-id 100;
      interface {
          ge-0/0/10.0;
          ge-0/0/0.0;
      }
```

```
}
erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/20.0;
        ge-0/0/10.0;
        ge-0/0/0.0;
    }
}
erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/20.0;
        ge-0/0/10.0;
        ge-0/0/0.0;
    }
}
```

In configuration mode, check your interface configurations by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show interfaces
  ge-0/0/0 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
  }
  ge-0/0/10 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
  }
  ge-0/0/20 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
  }
```

If you are finished configuring the device, enter **commit** in configuration mode.

Configuring ERPS on Hairtail

CLI Quick Configuration

To quickly configure Hairtail, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols rstp interface ge-0/0/10 disable
set protocols rstp interface ge-0/0/20 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
```

```
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel
  ge-0/0/10.0
```

Step-by-Step Procedure

To configure ERPS on Hairtail:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/10 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control VLAN erp-control-vlan-1 for the node ring erp1:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan erp-control-vlan-1
   ```

4. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel erp-data-1
   user@switch# set data-channel erp-data-2
   ```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/20.0 and indicate that it connects to a ring protection link:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set east-interface control-channel ge-0/0/20.0
   ```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/10.0 and indicate that it connects to a ring protection link:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set west-interface control-channel ge-0/0/10.0
   ```

Results

In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
```

```
rstp {
    interface ge-0/0/10.0 {
        disable;
    }
    interface ge-0/0/20.0 {
        disable;
    }
}
protection-group {
    ethernet-ring erp1 {
        east-interface {
            control-channel {
                ge-0/0/20.0;
            }
        }
        west-interface {
            control-channel {
            ge-0/0/10.0;
            }
        }
        control-vlan erp-control-vlan-1;
        data-channel {
            vlan [ 101-102 ];
        }
    }
}
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show vlans
 erp-control-vlan-1 {
    vlan-id 100;
    interface {
        ge-0/0/20.0;
        ge-0/0/10.0;
    }
 }
 erp-data-1 {
    vlan-id 101;
    interface {
        ge-0/0/0.0;
        ge-0/0/20.0;
        ge-0/0/10.0;
    }
 }
 erp-data-2 {
    vlan-id 102;
    interface {
        ge-0/0/0.0;
        ge-0/0/20.0;
        ge-0/0/10.0;
    }
 }
```

In configuration mode, check your interface configurations by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show interfaces
  ge-0/0/0 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
  }
  ge-0/0/10 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
  ge-0/0/20 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
```

If you are finished configuring the device, enter **commit** in configuration mode.

Configuring ERPS on Jas6-esc

CLI Quick Configuration

To quickly configure Jas6-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols rstp interface ge-0/0/30 disable
set protocols rstp interface ge-0/0/20 disable
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-1
set protocols protection-group ethernet-ring erp1 data-channel erp-data-2
set protocols protection-group ethernet-ring erp1 control-vlan erp-control-vlan-1
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/30.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel
  ge-0/0/20.0
```

Step-by-Step Procedure

To configure ERPS on Jas6-esc:

1. Disable any spanning- tree protocols configured on the ERPS interfaces. RSTP is enabled in the default configuration, so this example shows disabling RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/30 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control VLAN erp-control-vlan-1 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan erp-control-vlan-1
```

4. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance.

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel erp-data-1
user@switch# set data-channel erp-data-2
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/30.0 :

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/30.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/20.0
```

Results In configuration mode, check your ERPS configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show protocols
 rstp {
     interface ge-0/0/20.0 {
         disable;
     }
     interface ge-0/0/30.0 {
         disable;
     }
 }
 protection-group {
     ethernet-ring erp1 {
         east-interface {
             control-channel {
                 ge-0/0/30.0;
             }
         }
         west-interface {
             control-channel {
                 ge-0/0/20.0;
             }
         }
         control-vlan erp-control-vlan-1;
         data-channel {
             vlan [ 101-102 ];
         }
     }
 }
```

In configuration mode, check your VLAN configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show vlans
  erp-control-vlan-1 {
      vlan-id 100;
      interface {
          ge-0/0/30.0;
          ge-0/0/20.0;
      }
  }
  erp-data-1 {
      vlan-id 101;
      interface {
          ge-0/0/0.0;
          ge-0/0/30.0;
          ge-0/0/20.0;
      }
  }
  erp-data-2 {
      vlan-id 102;
      interface {
          ge-0/0/0.0;
          ge-0/0/30.0;
          ge-0/0/20.0;
      }
  }
```

In configuration mode, check your interfaces configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@switch# show interfaces
  ge-0/0/0 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
  }
  ge-0/0/20 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
          }
      }
  }
  ge-0/0/30 {
      unit 0 {
          family ethernet-switching {
              port-mode trunk;
         }
      }
  }
```

Verification

Verify that ERPS is working correctly.

Verifying That ERPS Is Working Correctly

Purpose Verify that ERPS is working on the four EX switches that function as nodes in the ring topology.

Action Check the state of the ring links in the output of the **show protection-group ethernet-ring interface** command. When the ring is configured but not being used (no error exists on the data links), one ERP interface is forwarding traffic and one is discarding traffic. Discarding blocks the ring.

```
user@switch> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group erp1
Interface     Forward State  RPL End  Signal Failure    Admin State
ge-0/0/2.0    discarding     yes      clear             ready
ge-0/0/0.0    forwarding     no       clear             ready
```

To find out what has occurred since the last restart, check the RPS statistics for ring-blocked events. **NR** is a No Request ring block, which means that the switch is not blocking either of the two ERP interfaces. **NR-RB** is a No Request Ring Blocked event, which means that the switch is blocking one of its ERP interfaces and sending a packet out to notify the other switches.

```
user@switch> show protection-group ethernet-ring statistics
Ring Name Local SF Remote SF NR Event NR-RB Event
erp1       2          1            2            3
```

Meaning The **show protection-group ethernet-ring interface** command output from the RPL owner node indicates that one interface is forwarding traffic and one is discarding traffic, meaning that the ERP is ready but not active. If at least one interface in the ring is not forwarding, the ring is blocked and therefore ERP is working.

The **show protection-group ethernet-ring statistics** command output indicates that, since the last reboot, both local and remote signal failures have occurred (**Local SF** and **Remote SF**).

The **NR Event** count is 2, indicating that the NR state was entered into twice. **NR** stands for No Request. This means that the switch either originated NR PDUs or received an NR PDU from another switch and stopped blocking the interface to allow ERP to function.

The three **NR-RB** events indicate that on three occasions, this switch either sent out NR-RB PDUs or received NR-RB PDUs from another switch. This occurs when a network problem is resolved and the switch once again blocks the ERP link at one end.

Related Documentation

- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416
- Ethernet Ring Protection Switching Overview on page 407
- Understanding Ethernet Ring Protection Switching Functionality on page 408

Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS

You can configure Ethernet ring protection switching (ERPS) on connected EX Series or QFX Series switches to prevent fatal loops from disrupting a network. ERPS is similar to the Spanning Tree Protocol, but ERPS is more efficient because it is customized for ring topologies. You must connect and configure at least three switches to form a ring.

This example shows how to configure Ethernet ring protection switching on four switches with ELS support, connected to one another on a dedicated link in a ring topology. You can include different types of switches in an ERPS ring, including those with and without ELS support. If any of your EX Series switches runs software that does not support ELS, use these configuration directions: "Example: Configuring Ethernet Ring Protection Switching on EX Series Switches" on page 420. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

Requirements

This example uses the following hardware and software components:

- Four connected EX Series switches or QFX Series switches that support the Enhanced Layer 2 Software (ELS) to function as nodes in the ring topology. You could use any of these QFX Series switches: QFX5100, QFX5200, and QFX10000. This configuration also applies to EX Series switches that support the Enhanced Layer 2 Software (ELS) configuration style that runs on EX4300, EX4600, EX2300, and EX3400 switches.
- Junos OS Release 13.2X50-D10 or later for EX Series switches.
- Junos OS Release 14.1X53-D10 or later for QFX5100 switches. Junos OS Release 15.1X53-D30 or later for QFX5200, and QFX10000 switches.

Before you begin, be sure you have:

- Configured two trunk interfaces on each of the four switches. See Table 75 on page 422 for a list of the interface names used in this example.
- Configured a VLAN (with name **erp-control-vlan-1** and ID **100**) on all four switches and associated two network interfaces from each of the four switches with the VLAN. See *Configuring VLANs for the QFX Series* OR "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102. See Table 75 on page 422 for a list of the interface names used in this example.

- Configured two more VLANs (one with name **erp-data-1** and vlan ID **101** and a second vlan with the name **erp-data-2** and vlan ID **102**) on all four switches and associated both the east and west interfaces on each switch.

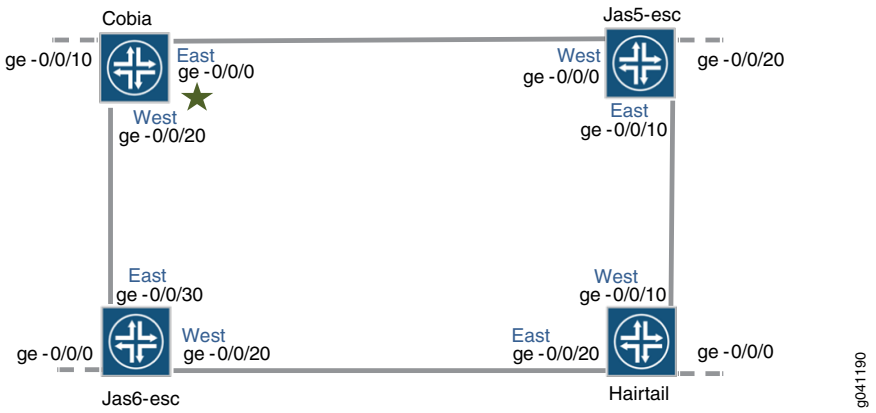
Overview and Topology

ERPS uses a dedicated physical link, including a control VLAN for trunk ports, between all of the switches to protect the active links. ERPS VLANs are all located on this link and are also blocked by default. When traffic between the switches is flowing with no problems, the active links take care of all traffic. Only if an error occurs on one of the data links would the ERPS control channel take over and start forwarding traffic.

NOTE: **Trunk ports on switches use a VLAN to create individual control channels for ERPS. When multiple ERPS instances are configured for a ring, there are multiple sets of ring protection links (RPLs) and RPL owners on the ERPS link, and a different channel is blocked for each instance. Nontrunk ports use the physical link as the control channel and protocol data units (PDUs) are untagged, with no VLAN information in the packet.**

This example creates one protection ring (called a node ring) named erp1 on four switches connected in a ring by trunk ports as shown in Figure 27 on page 422. Because the links are trunk ports, VLAN 100 is used for erp1 traffic. The east interface of each switch is connected with the west interface of an adjacent switch. Cobia is the RPL owner, with interface ge-0/0/0 configured as an RPL end interface. The interface ge-0/0/0 of Jas5-esc is configured as the RPL neighbor interface. In the idle state, the RPL end blocks the control VLAN and data channel VLAN for this particular ERP instance—the blocked port on Cobia is marked with a star in Figure 27 on page 422.

Figure 28: Ethernet Ring Protection Switching Example



In this example, we configure the four switches with the interfaces indicated in both Figure 27 on page 422 and Table 75 on page 422.

Table 76: Components to Configure for This Example

Interfaces	Cobia	Jas5-esc	Jas6-esc	Hairtail
East	ge-0/0/0	ge-0/0/10	ge-0/0/30	ge-0/0/20
West	ge-0/0/20	ge-0/0/0	ge-0/0/20	ge-0/0/10
Third	ge-0/0/10	ge-0/0/20	ge-0/0/0	ge-0/0/0

Configuration

Configuring ERPS on Cobia, the RPL Owner Node

CLI Quick Configuration

To quickly configure Cobia, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning-tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements in this example vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/0 disable
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/20 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/0
Junos OS release prior to 15.1: delete rstp interface ge-0/0/20
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 ring-protection-link-owner
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 east-interface ring-protection-link-end
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan
  100
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan
  100
```

Step-by-Step Procedure

To configure ERPS on Cobia:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/0 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

   If you are running a Junos release prior to 15.1, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/0
   user@switch# delete rstp interface ge-0/0/20
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Designate Cobia as the RPL owner node:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set ring-protection-link-owner
   ```

4. Configure the VLANs 101 and 102 as data channels:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel 101
   user@switch# set data-channel 102
   ```

5. Configure the control vlan 100 for this ERPS instance on the trunk interface:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan 100
   ```

6. Configure the east interface of the node ring erp1 with control channel ge-0/0/0.0 and indicate that this particular ring protection link ends here:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set east-interface control-channel ge-0/0/0.0
   user@switch# set east-interface ring-protection-link-end
   ```

7. Configure the west interface of the node ring erp1 with control channel ge-0/0/20.0 :

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set west-interface control-channel ge-0/0/20.0
   ```

8. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign 100 as the control VLAN on both interfaces:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel
```

Configuring ERPS on Jas5-esc

CLI Quick Configuration

To quickly configure Jas5-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning tree is disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements will vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/10 disable
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/0 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/10
Junos OS release prior to 15.1: delete rstp interface ge-0/0/0
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/10.0
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan
  100 ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel
  ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan
  100
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan
  100
```

Step-by-Step Procedure

To configure ERPS on Jas5-esc:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

```
[edit protocols]
user@switch# set rstp interface ge-0/0/10 disable
```

```
user@switch# set rstp interface ge-0/0/0 disable
```

If you are running a Junos release prior to 15.1, disable any version of spanning-tree protocol with these commands. To disable RSTP:

```
[edit protocols]
user@switch# delete rstp interface ge-0/0/10
user@switch# delete rstp interface ge-0/0/0
```

2. Create a node ring named erp1:

```
[edit protocols]
user@switch# set protection-group ethernet-ring erp1
```

3. Configure two data channels named erp-data-1 and erp-data-2 to define a set of VLAN IDs that belong to a ring instance.

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel vlan 101
user@switch# set data-channel vlan 102
```

4. Configure a control VLAN with ID 100 for the node ring erp1:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set control-vlan 100
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/10.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/10.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/0.0 vlan 100:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/0.0
```

7. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign vlan # 100 as the control VLAN:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel vlan 100
```

Configuring ERPS on Hairtail

CLI Quick Configuration

To quickly configure Hairtail, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning tree is disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements will vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/10 disable
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/20 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/10
Junos OS release prior to 15.1: delete rstp interface ge-0/0/20
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
Set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/0.0
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/20.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan
  100
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan
  100
```

Step-by-Step Procedure

To configure ERPS on Hairtail:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/10 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

   If you are running a Junos release prior to 15.1, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/10
   user@switch# delete rstp interface ge-0/0/20
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control vlan 100 for the node ring erp1:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan 100
   ```

4. Configure two data channels numbered 101 and 102 to define a set of VLAN IDs that belong to a ring instance:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set data-channel vlan 101
user@switch# set data-channel vlan 102
```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/20.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set east-interface control-channel ge-0/0/20.0
```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/10.0:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel ge-0/0/10.0
```

7. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign 100 as the control VLAN:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel vlan 100
```

Configuring ERPS on Jas6-esc

CLI Quick Configuration

To quickly configure Jas6-esc, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

ERPS cannot be configured on an interface if any spanning tree protocol is configured. (RSTP is configured by default.) Therefore, in this example, RSTP is disabled on each ring port before configuring ERPS. Spanning tree is disabled two different ways, depending on which version of the Junos OS you are running. Therefore, the first two statements will vary: Junos release 15.1 or later uses one command to turn off RSTP and Junos releases prior to 15.1 uses another command.

```
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/30 disable
Junos OS release 15.1 or later: set protocols rstp interface ge-0/0/20 disable
Junos OS release prior to 15.1: delete rstp interface ge-0/0/30
Junos OS release prior to 15.1: delete rstp interface ge-0/0/20
set protocols protection-group ethernet-ring erp1
set protocols protection-group ethernet-ring erp1 data-channel 101
set protocols protection-group ethernet-ring erp1 data-channel 102
set protocols protection-group ethernet-ring erp1 control-vlan 100
set protocols protection-group ethernet-ring erp1 east-interface control-channel
  ge-0/0/30.0
set protocols protection-group ethernet-ring erp1 west-interface control-channel
  ge-0/0/20.0
```

```
set protocols protection-group ethernet-ring erp1 west-interface control-channel vlan
  100
set protocols protection-group ethernet-ring erp1 east-interface control-channel vlan
  100
```

Step-by-Step Procedure

To configure ERPS on Jas6-esc:

1. Disable any spanning-tree protocol currently configured on the ERPS interfaces. RSTP, VSTP, and MSTP are all available spanning-tree protocols. RSTP is enabled in the default configuration, so this example shows disabling RSTP. Spanning-tree protocols are disabled two different ways, depending on which version of the Junos OS you are running.

   If you are running Junos release 15.1 or later, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# set rstp interface ge-0/0/30 disable
   user@switch# set rstp interface ge-0/0/20 disable
   ```

   If you are running a Junos release prior to 15.1, disable any spanning-tree protocol with these commands. To disable RSTP:

   ```
   [edit protocols]
   user@switch# delete rstp interface ge-0/0/30
   user@switch# delete rstp interface ge-0/0/20
   ```

2. Create a node ring named erp1:

   ```
   [edit protocols]
   user@switch# set protection-group ethernet-ring erp1
   ```

3. Configure the control vlan 100 for the node ring erp1:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set control-vlan 100
   ```

4. Configure two data channels numbered 101 and 102 to define VLAN IDs that belong to a ring instance.

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set data-channel 101
   user@switch# set data-channel 102
   ```

5. Configure the east interface of the node ring erp1 with the control channel ge-0/0/30.0 :

   ```
   [edit protocols protection-group ethernet-ring erp1]
   user@switch# set east-interface control-channel ge-0/0/30.0
   ```

6. Configure the west interface of the node ring erp1 with the control channel ge-0/0/20.0:

   ```
   [edit protocols protection-group ethernet-ring erp1]
   ```

```
user@switch# set west-interface control-channel ge-0/0/20.0
```

7. Every ring instance on a trunk port has one control VLAN in which ERP packets traverse. The control VLAN also controls data VLANs, if any are configured. Assign vlan number 100 as the control VLAN:

```
[edit protocols protection-group ethernet-ring erp1]
user@switch# set west-interface control-channel vlan 100
user@switch# set east-interface control-channel vlan 100
```

Related Documentation

- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416
- Ethernet Ring Protection Switching Overview on page 407
- Understanding Ethernet Ring Protection Switching Functionality on page 408

CHAPTER 18

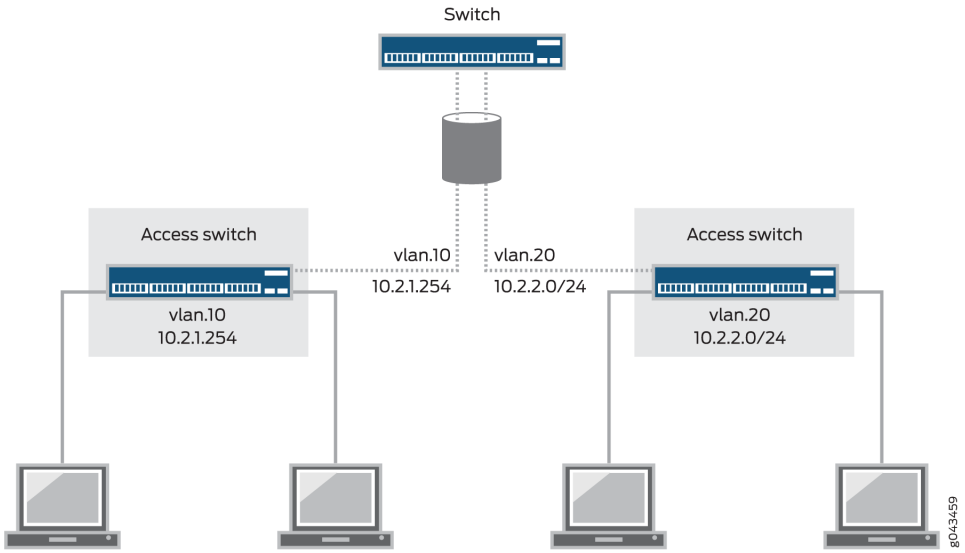
Configuring Integrated Routing and Bridging (IRB)

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.

Figure 29 on page 446 illustrates a switch routing VLAN traffic between two access layer switches using one of these interfaces.

Figure 29: An IRB Interface or RVI on a Switch Providing Routing Between Two Access Switches



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. Packets arriving on a Layer 2 interface that are destined for the device's MAC address are classified as Layer 3 traffic while packets that are not destined for the device's MAC address are classified as Layer 2 traffic. Packets destined for the device's MAC address are sent to the IRB interface. Packets from the device's routing engine are sent out the IRB interface.

NOTE: If you specify a VLAN identifier list in the VLAN configuration, you cannot configure an IRB interface for the VLAN.

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

IRB interfaces supporting the Enhanced Layer 2 Software (ELS) configuration style and RVIs that support non-ELS switches provide the same functionality. Where the functionality for both features is the same, this topic uses the term *these interfaces* to refer collectively to both IRB interfaces and RVIs. Where differences exist between the two features, this topic calls out the IRB interfaces and RVIs separately.

Table 77 on page 447 shows values you might use when configuring an IRB:

Table 77: 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**
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 77 on page 447 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 78 on page 447 shows the number of IRBs/RVIs that each QFX platform supports.

Table 78: Number of Supported IRBs/RVIs by Platform

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

RB Interfaces on SRX Series Devices

On SRX1400, SRX1500, SRX3400, SRX3600, SRX4100, SRX4200, SRX4600, SRX5600, and SRX5800 devices, Juniper supports an IRB interface that allows you to terminate management connections in transparent mode. However, you cannot route traffic on that interface or terminate IPsec VPNs. (Platform support depends on the Junos OS release in your installation.)

NOTE: You can configure only one IRB logical interface for each VLAN.

On SRX300, SRX320, SRX340, SRX345 devices, and SRX550M on the IRB interface, the following features are not supported:

- IS-IS (family ISO)
- Encapsulations (Ether CCC, VLAN CCC, VPLS, PPPoE, and so on) on VLAN interfaces
- CLNS
- DVMRP
- VLAN interface MAC change
- G-ARP
- Change VLAN-Id for VLAN interface

NOTE: Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, interface statistics are supported on the IRB logical interface for SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

To verify the IRB logical interface statistics, enter the **show interfaces irb.<index> extensive** and **show interfaces irb.<index>statistics** commands.

When Should I Use an IRB Interface or RVI?

Configure an IRB interface or an RVI for a VLAN if you need to:

- Allow traffic to be routed between VLANs.
- Provide Layer 3 IP connectivity to the switch.
- Monitor individual VLANs for billing purposes. Service providers often need to monitor traffic for this purpose, but this capability can be useful for enterprises where various groups share the cost of the network.

How Does an IRB Interface or RVI Work?

For an IRB interface, the switch provides the name irb, and for an RVI, the switch provides the name vlan. Like all Layer 3 interfaces, these interfaces require a logical unit number with an IP address assigned to it. In fact, to be useful, the implementation of these

interfaces in an enterprise with multiple VLANs requires 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 interfaces 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.

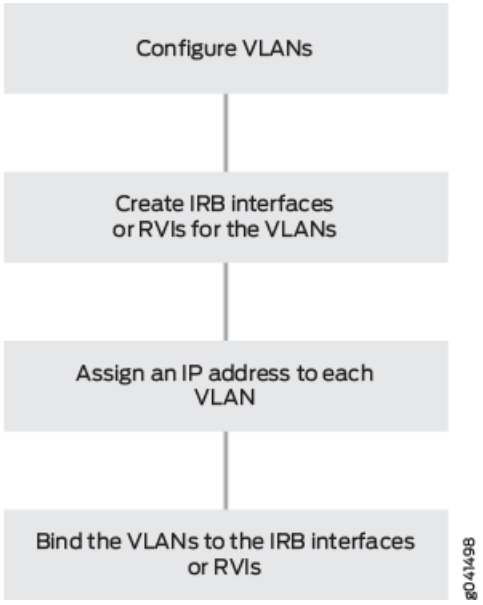
The interface on the switch detects both MAC addresses and IP addresses, then routes data to other Layer 3 interfaces on routers or other switches. These interfaces detect both IPv4 and IPv6 unicast and multicast virtual routing and forwarding (VRF) traffic. Each logical interface can belong to only one routing instance and is further subdivided into logical interfaces, each with a logical interface number appended as a suffix to the names irb and vlan—for example, irb.10 and vlan.10.

Creating an IRB Interface or RVI

You create an IRB logical interface in a similar manner as a Layer 3 interface, but the IRB interface does not support traffic forwarding or routing. The IRB interface cannot be assigned to a security zone; however, you can configure certain services on a per-zone basis to allow host-inbound traffic for management of the device. This allows you to control the type of traffic that can reach the device from interfaces bound to a specific zone.

There are four basic steps in creating an IRB interface or RVI as shown in Figure 30 on page 449.

Figure 30: Creating an IRB Interface or RVI



The following explanations correspond to the four steps for creating a VLAN, as depicted in Figure 30 on page 449.

- Configure VLANs—Virtual LANs are groups of hosts that communicate as if they were attached to the same broadcast stream. VLANs are created with software and do not require a physical router to forward traffic. VLANs are Layer 2 constructs.
- Create IRB interfaces or RVIs for the VLANs—The switch's IRB interfaces and RVIs use Layer 3 logical interfaces (unlike routers, which can use either physical or logical interfaces).
- Assign an IP address to each VLAN—An IRB interface or RVI cannot be activated unless it is associated with a physical interface.
- Bind the VLANs to the logical interfaces—There is a one-to-one mapping between a VLAN and an IRB interface or RVI, which means that only one of these interfaces can be mapped to a VLAN.

For specific instructions for creating an IRB interface, see "Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure)" on page 456, and for an RVI, see "Configuring Routed VLAN Interfaces on Switches (CLI Procedure)" on page 365.

Viewing IRB Interface and RVI Statistics

Some switches automatically track IRB interface and RVI traffic statistics. Other switches allow you to configure tracking. Table 79 on page 450 illustrates the IRB interface- and RVI-tracking capability on various switches.

Table 79: Tracking IRB Interface and RVI Usage

Switch	Input (ingress)	Output (Egress)
EX4300	Automatic	Automatic
EX3200, EX4200	Automatic	–
EX8200	Configurable	Automatic
EX2200, EX3300, EX4500, EX6200	–	–

You can view input (ingress) and output (egress) totals with the following commands:

- For IRB interfaces, use the **show interfaces irb extensive** command. Look at the input and output values in the Transit Statistics field for IRB interface activity values.
- For RVI, use the **show interfaces vlan extensive** command. Look at the input and output values in the Logical Interface Transit Statistics field for RVI activity values.

IRB Interfaces and RVI Functions and Other Technologies

IRB interfaces and RVIs are similar to switch virtual interfaces (SVIs) and bridge-group virtual interfaces (BVIs), which are supported on other vendors' devices. They can also be combined with other functions:

- VRF is often used in conjunction with Layer 3 subinterfaces, allowing traffic on a single physical interface to be differentiated and associated with multiple virtual routers. For

more information about VRF, see "Understanding Virtual Routing Instances on EX Series Switches" on page 487.

- For redundancy, you can combine an IRB interface or RVI with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments. For more information about VRRP, see *Understanding VRRP*.

Related Documentation

- Example: Configuring Routing Between VLANs on One Switch Using an IRB Interface on page 462
- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring an IRB Interface on a Security Device on page 452
- Understanding VLANs on Security Devices on page 380
- Example: Configuring VLANs on Security Devices on page 382

Using an IRB Interface in a Private VLAN on a Switch

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. 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.

Just like regular VLANs, PVLANs are isolated at Layer 2 and normally require that a Layer 3 device be used if you want to route traffic. Starting with Junos OS 14.1X53-D30, you can use an integrated routing and bridging (IRB) interface to route Layer 3 traffic between devices connected to a PVLAN. Using an IRB interface in this way can also allow the devices in the PVLAN to communicate at Layer 3 with devices outside the PVLAN.

- Configuring an IRB Interface in a Private VLAN on page 451
- IRB Interface Limitation in a PVLAN on page 452

Configuring an IRB Interface in a Private VLAN

Use the following guidelines when configuring an IRB interface in a PVLAN:

- You can create only one IRB interface in a PVLAN, regardless of how many switches participate in the PVLAN.
- The IRB interface must be a member of the primary VLAN in the PVLAN.
- Each host device that you want to connect at Layer 3 must use the IP address of the IRB as its default gateway address.
- · Because the host devices are isolated at Layer 2, you must configure the following statement for the IRB interface to allow ARP resolution to occur:

set interfaces irb unit *unit-number* proxy-arp unrestricted

IRB Interface Limitation in a PVLAN

If your PVLAN includes multiple switches, an issue can occur if the Ethernet switching table is cleared on a switch that does not have an IRB interface. If a Layer 3 packet transits the switch before its destination MAC address is learned again, it is broadcast to all the Layer 3 hosts connected to the PVLAN.

Related Documentation

- Understanding Private VLANs on page 226
- Configuring IRB Interfaces on Switches on page 454
- Creating a Private VLAN on a Single QFX Switch on page 269
- Example: Configuring a Private VLAN Spanning Multiple Switches With an IRB Interface on page 310

Example: Configuring an IRB Interface on a Security Device

This example shows how to configure an IRB interface so it can act as a Layer 3 routing interface for a VLAN.

Requirements

Before you begin, configure a VLAN with a single VLAN identifier. See "Example: Configuring VLANs on Security Devices" on page 382.

Overview

In this example, you configure the IRB logical interface unit 0 with the family type inet and IP address 10.1.1.1/24, and then reference the IRB interface irb.10 in the vlan10 configuration. Then you enable Web authentication on the IRB interface and activate the webserver on the device.

NOTE: To complete the Web authentication configuration, you must perform the following tasks:

- Define the access profile and password for a Web authentication client.
- Define the security policy that enables Web authentication for the client.

Either a local database or an external authentication server can be used as the Web authentication server.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-1/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members 10
set interface irb unit 0 family inet address 10.1.1.1/24 web-authentication http
set vlans vlan10 vlan-id 10
set vlans vlan10 l3-interface irb.10
set system services web-management http
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an IRB interface:

1. Create a Layer 2 trunk interface.

   ```
   [edit]
   user@host# set interfaces ge-1/0/0 unit 0 family ethernet-switching interface-mode
     trunk
   user@host# set interfaces ge-1/0/0 unit 0 family ethernet-switching vlan members
     10
   ```

2. Create an IRB logical interface.

   ```
   [edit]
   user@host# set interface irb unit 0 family inet address 10.1.1.1/24 web-authentication
     http
   ```

3. Create a Layer 2 VLAN.

   ```
   [edit]
   user@host# set vlans vlan10 vlan-id 10
   ```

4. Associate the IRB interface with the VLAN.

   ```
   [edit]
   user@host# set vlans vlan10 l3-interface irb.10
   ```

5. Activate the webserver.

   ```
   [edit]
   user@host# set system services web-management http
   ```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interface irb** , and **show vlans** commands.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Understanding Integrated Routing and Bridging on page 445
- Example: Configuring Layer 2 Security Zones on page 652
- Understanding VLANs on Security Devices on page 380

Configuring IRB Interfaces on Switches

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).

NOTE: When you upgrade from Junos OS Release 15.1X53 to Junos OS Release 17.3R1, you must define an IRB interface at both the **[edit vlans l3-interface]** and **[edit interfaces irb]** hierarchies, otherwise there will be a commit error.

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 10.0.0.X/8
```

Where the value of X can be any number between the range 1 to 254.

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     10.0.0.0/8


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 445

Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure)

Integrated routing and bridging (IRB) interfaces allow a switch to recognize packets that are being sent to local addresses so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated.

An interface named irb functions as a logical router on which you can configure a Layer 3 logical interface for each virtual LAN (VLAN). For redundancy, you can combine an IRB interface with implementations of the Virtual Router Redundancy Protocol (VRRP) in both bridging and virtual private LAN service (VPLS) environments.

Jumbo frames of up to 9216 bytes are supported on an IRB interface. To route jumbo data packets on the IRB interface, you must configure the jumbo MTU size on the member physical interfaces of the VLAN that you have associated with the IRB interface, as well as on the IRB interface itself (the interface named irb).

CAUTION: Setting or deleting the jumbo MTU size on the IRB interface (the interface named irb) while the switch is transmitting packets might result in dropped packets.

To configure the IRB interface:

1. Create a Layer 2 VLAN by assigning it a name and a VLAN ID:

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

2. Assign an interface to the VLAN by naming the VLAN as a trunk member on the logical interface, thereby making the interface part of the VLAN's broadcast domain:

   ```
   [edit]
   user@switch# set interfaces interface-name unit logical-unit-number family ethernet-switching
   vlan members vlan-name
   ```

3. Create a logical Layer 3 IRB interface (its name will be irb.*logical-interface-number*, where the value for *logical-interface-number* is the value you supplied for *vlan-id* in Step 1; in the following command, it is the *logical-unit-number*) on a subnet for the VLAN's broadcast domain:

   ```
   [edit]
   user@switch# set interfaces irb unit logical-unit-number family inet address inet-address
   ```

4. Link the Layer 2 VLAN to the logical Layer 3 IRB interface:

   ```
   [edit]
   user@switch# set vlans vlan-name l3-interface irb.logical-interface-number
   ```

NOTE: Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple Layer 2 VLANs. Within a VLAN, traffic is switched, while across VLANs, traffic is routed.

Related Documentation

- Verifying Integrated Routing and Bridging Interface Status and Statistics on EX Series Switches on page 483
- Understanding Integrated Routing and Bridging on page 445

Example: Configuring IRB and VLAN with Members Across Two Nodes on a Security Device

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example shows the configuration of integrated routing and bridging (IRB) and configuration of a VLAN with members across node 0 and node 1.

Configuration

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-7/0/5 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members vlan100
set interfaces irb unit 100 family inet address 192.0.2.100/24
set vlans vlan100 vlan-id 100
set vlans vlan100 l3-interface irb.100
```

Step-by-Step Procedure

To configure IRB and a VLAN:

1. Configure Ethernet switching on the node0 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode
  access
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode
  access
```

2. Configure Ethernet switching on the node1 interface.

```
{primary:node0} [edit]
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching interface-mode
  trunk
```

3. Create VLAN vlan100 with vlan-id 100.

```
{primary:node0} [edit]
user@host# set vlans vlan100 vlan-id 100
```

4. Add interfaces from both nodes to the VLAN.

```
{primary:node0} [edit]
user@host# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members
  vlan100
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members
  vlan100
user@host# set interfaces ge-7/0/5 unit 0 family ethernet-switching vlan members
  vlan100
```

5. Create an IRB logical interface.

```
user@host# set interfaces irb unit 100 family inet address 192.0.2.100/24
```

6. Associate an IRB interface with the VLAN.

```
user@host# set vlans vlan100 l3-interface irb.100
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results From configuration mode, confirm your configuration by entering the **show vlans** and **show interfaces** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct the configuration.

```
[edit]
user@host# show vlans
vlan100 {
   vlan-id 100;
   l3-interface irb.100;
}
[edit]
user@host# show interfaces
```

```
ge-0/0/3 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members vlan100;
      }
    }
  }
}
ge-0/0/4 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan {
        members vlan100;
      }
    }
  }
}
ge-7/0/5 {
  unit 0 {
    family ethernet-switching {
      interface-mode trunk;
      vlan {
        members vlan100;
      }
    }
  }
}
irb {
  unit 100 {
    family inet {
      address 192.0.2.100/24;
    }
  }
}
```

Verification

Verifying VLAN and IRB

Purpose Verify that the configurations of VLAN and IRB are working properly.

Action From operational mode, enter the **show interfaces terse ge-0/0/3** command to view the node 0 interface.

```
user@host> show interfaces terse ge-0/0/3
Interface               Admin Link Proto    Local                 Remote
ge-0/0/3                up    up
ge-0/0/3.0              up    up   eth-switch
```

From operational mode, enter the **show interfaces terse ge-0/0/4** command to view the node 0 interface.

```
user@host> show interfaces terse ge-0/0/4
Interface               Admin Link Proto    Local                 Remote
ge-0/0/4                up    up
ge-0/0/4.0              up    up   eth-switch
```

From operational mode, enter the **show interfaces terse ge-7/0/5** command to view the node1 interface.

```
user@host> show interfaces terse ge-7/0/5
Interface               Admin Link Proto    Local                 Remote
ge-7/0/5                up    up
ge-7/0/5.0              up    up   eth-switch
```

From operational mode, enter the **show vlans** command to view the VLAN interface.

```
user@host> show vlans
Routing instance      VLAN name     Tag      Interfaces
default-switch        default        1
default-switch        vlan100        100     ge-0/0/3.0*
                                             ge-0/0/4.0*
                                             ge-7/0/5.0*
```

From operational mode, enter the **show ethernet-switching interface** command to view the information about 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,
                        MMAS - Mac-move action shutdown,  AS - Autostate-exclude
 enabled,
                         SCTL - shutdown by Storm-control )

Logical         Vlan                  TAG   MAC    STP         Logical
 Tagging
interface       members                     limit  state       interface flags
ge-0/0/3.0                                  16383              DN
 untagged
                vlan100               100   1024   Discarding
 untagged
ge-0/0/4.0                                  16383              DN
 untagged
                vlan100               100   1024   Discarding
 untagged
ge-7/0/5.0                                  16383              DN
 tagged
                vlan100               100   1024   Discarding
 tagged
```

Meaning The output shows the VLAN and IRB are configured and working fine.

Related Documentation

- Example: Configuring an IRB Interface

Configuring Integrated Routing and Bridging for VLANs

Integrated routing and bridging (IRB) provides simultaneous support for Layer 2 bridging and Layer 3 routing on the same interface. IRB enables you to route packets to another routed interface or to another VLAN that has an IRB interface configured. You configure a logical routing interface by specifying **irb** as an interface name at the **[edit interfaces]** hierarchy level and including that interface in the VLAN.

NOTE: You can include only one Layer 3 interface in a VLAN.

To configure a VLAN with IRB support, include the following statements:

```
[edit]
vlans {
  vlan-name {
    domain-type bridge;
    interface interface-name;
    l3-interface (VLAN) interface-name;
    vlan-id (none | number);
    vlan-tags outer number inner number;
  }
}
```

For each VLAN that you configure, specify a ***vlan-name***. You must also specify the value **bridge** for the **domain-type** statement.

For the **vlan-id** statement, you can specify either a valid VLAN identifier or the **none** option.

NOTE: If you configure a Layer 3 interface to support IRB in a VLAN, you cannot use the **all** option for the **vlan-id** statement.

The **vlan-tags** statement enables you to specify a pair of VLAN identifiers; an **outer** tag and an **inner** tag.

NOTE: For a single VLAN, you can include either the **vlan-id** statement or the **vlan-tags** statement, but not both.

To include one or more logical interfaces in the VLAN, specify the ***interface-name*** for each Ethernet interface to include that you configured at the **[edit interfaces]** hierarchy level.

NOTE: A maximum of 4096 active logical interfaces are supported for a VLAN or on each mesh group in a VPLS routing instance configured for Layer 2 bridging.

To associate a Layer 3 interface with a VLAN, include the **l3-interface** ***interface-name*** statement and specify an ***interface-name*** you configured at the **[edit interfaces irb]** hierarchy level. You can configure only one Layer 3 interface for each VLAN.

IRB interfaces are supported for multicast snooping.

In multihomed VPLS configurations, you can configure VPLS to keep a VPLS connection up if only an IRB interface is available by configuring the **irb** option for the **connectivity-type** statement at the **[edit routing-instances** ***routing-instance-name*** **protocols vpls]** hierarchy level. The **connectivity-type** statement has the **ce** and **irb** options. The **ce** option is the default and specifies that a CE interface is required to maintain the VPLS connection. By default, if only an IRB interface is available, the VPLS connection is brought down.

NOTE: When you configure IRB interfaces in more than one logical system on a device, all of the IRB logical interfaces share the same MAC address.

Example: Configuring Routing Between VLANs on One Switch Using an IRB Interface

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

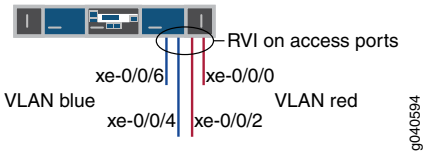
This example uses the following hardware and software components:

- 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 31 on page 463.

Figure 31: 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 80 on page 463 lists the components of the sample topology.

Table 80: 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
```

Configuration Results

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 467
- Verifying That Traffic Can Be Routed Between the Two VLANs on page 467

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 445
- irb (Interfaces) on page 1003
- l3-interface on page 1014

Excluding an IRB Interface from State Calculations on a QFX Series Switch

IRB interfaces are used to bind specific VLANs to Layer 3 interfaces, enabling a switch to forward packets between those VLANs— without having to configure another device, such as a router, to connect VLANs. Because an IRB interface often has multiple ports in a single VLAN, the state calculation for a VLAN member might include a port that is down, possibly resulting in traffic loss.

Starting with Junos OS Release 14.1X53-D40 and Junos OS Release 17.3R1 on QFX5100 switches, this feature enables you to exclude a trunk or access interface from the state calculation, which means that as soon as the port assigned to a member VLAN goes down, the IRB interface for the VLAN is also marked as down. In a typical scenario, one port on the interface is assigned to a single VLAN, while a second port on that interface is assigned to a trunk interface that carries traffic between multiple VLANs. A third port is often also assigned to an access interface to connect the VLAN to network devices.

Before you begin:

- Configure VLANs
- Configure IRB interfaces for the VLANs.

  For more information about configuring IRB interfaces, see “Example: Configuring Routing Between VLANs on One Switch Using an IRB Interface” on page 462.

To exclude an access or 802.1Q trunk interface from the state calculations for an IRB interface:

1. Configure a trunk or access interface.

   ```
   [edit interfaces interface-name]
   user@switch# set unit logical-unit-number family ethernet-switching port-mode
     (access | trunk)
   ```

   For example, configure interface xe-0/1/0.0 as a trunk interface:

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

2. Assign VLAN members to the access or trunk interface.

   ```
   [edit interfaces interface-name unit logical-unit-number ethernet-switching]
   user@switch# set vlan members [ (all | names | vlan-ids) ]
   ```

   For example, assign all VLAN members configured on the device to the trunk interface xe-0/1/0:

   ```
   [edit interfaces xe-0/1/0 unit 0 ethernet-switchg]
   user@switch# set vlan members all
   ```

3. Exclude an access or trunk interface from state calculations for the IRB interfaces for member VLANs.

   ```
   [edit interfaces interface-name ether-options]
   user@switch# set autostate-exclude
   ```

   For example, exclude the trunk interface xe-0/1/0 from state calculations for the IRB interfaces for member VLANs:

   ```
   [edit interfaces xe-0/1/0]
   user@switch# set autostate-exclude
   ```

4. To confirm your configuration, from configuration mode, enter the **show interfaces xe-0/1/0** command. If your output does not display the intended configuration, repeat steps 1 through 4 to correct the configuration.

   ```
   user@switch# show interfaces xe-0/1/0
   ether-options {
     autostate-exclude;
   }
   unit 0 {
     family ethernet-switching {
       port-mode trunk;
       vlan {
         members all;
       }
     }
   }
   ```

5. After you commit the configuration, issue the **show ethernet-switching interface xe-0/1/0.0** to verify that the logical interface is enabled with **autostate-exclude**.

```
user@switch> show ethernet-switching interface xe-0/1/0.0
Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop,
                         LH - MAC limit hit, DN - interface down,
                         SCTL - shutdown by Storm-control,
                         MMAS - Mac-move action shutdown, AS - Autostate-exclude enabled)

Logical          Vlan          TAG     MAC         STP         Logical           Tagging
interface        members               limit       state       interface flags
xe-0/1/0.0                             294912                   AS                untagged
                 vlan_100      100     294912      Forwarding                    untagged
```

The **AS** in the **Logical interface flags** field indicates that **autostate-exclude** is enabled and that this interface will be excluded from the state calculations for the IRB interfaces for the member VLANs.

Release History Table

Release	Description
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40 and Junos OS Release 17.3R1 on QFX5100 switches, this feature enables you to exclude a trunk or access interface from the state calculation, which means that as soon as the port assigned to a member VLAN goes down, the IRB interface for the VLAN is also marked as down.

Related Documentation

- Understanding Integrated Routing and Bridging on page 445

Example: Configuring IRB Interfaces on QFX5100 Switches over an MPLS Core Network

Starting with Junos OS Release 14.1X53-D40 and Junos OS Release 17.1R1, QFX5100 switches support integrated routing and bridging (IRB) interfaces over an MPLS core network. An IRB interface is a logical Layer 3 VLAN interface used to route traffic between VLANs.

By definition, VLANs divide a LAN's broadcast environment into isolated virtual broadcast domains, thereby limiting the amount of traffic flowing across the entire LAN and reducing the possible number of collisions and packet retransmissions within the LAN. To forward packets between different VLANs, you traditionally needed a router that connects the VLANs. However, using the Junos OS you can accomplish this inter-VLAN forwarding without using a router by simply configuring an IRB interface on the switch.

The IRB interface functions as a logical switch on which you can configure a Layer 3 logical interface for each VLAN. The switch relies on its Layer 3 capabilities to provide this basic routing between VLANs. With an IRB interface, you can configure label-switched paths (LSPs) to enable the switch to recognize which packets are being sent to local addresses, so that they are bridged (switched) whenever possible and are routed only when necessary. Whenever packets can be switched instead of routed, several layers of processing are eliminated.

This example shows how to configure an IRB interface over an MPLS core network using QFX5100 switches.

Requirements

This example uses the following hardware and software components:

- Three QFX5100 switches
- Junos OS Release 14.1X53-D40 or later

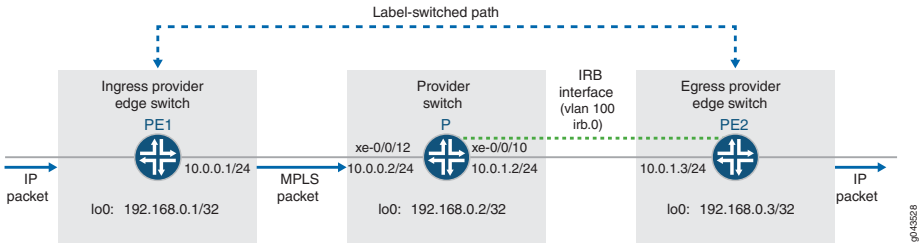
Before you begin, be sure you have:

- An understanding of IRB concepts. See "Understanding Integrated Routing and Bridging" on page 445 for an overview of IRB.
- The required ternary content addressable memory (TCAM) space available on the switch. TCAM rules must be observed while configuring and implementing IRBs. For detailed information, see *MPLS Limitations on QFX Series and EX4600 Switches*.

Overview and Topology

Figure 32 on page 471 illustrates a sample topology for configuring IRB over an MPLS core network. In this example, an LSP is established between the ingress provider edge switch (PE1) and the provider edge egress switch (PE2). An IRB Layer 3 interface (irb.0) is configured on switches P and PE2, and associated to VLAN 100. In this configuration, the P switch replaces (swaps) the label at the top of the label stack with a new label, adds the VLAN identifier 100 to the MPLS packet, and then sends the packet out the IRB interface. PE2 receives this vlan-tagged MPLS packet, removes (pops) the label from the top of the label stack, performs a regular IP route lookup, and then forwards the packet with its IP header to the next-hop address.

Figure 32: IRB Topology over an MPLS Core Network



Configuration

To configure the topology in this example, perform these tasks:

Configuring the Local Ingress PE Switch

CLI Quick Configuration

To quickly configure the local ingress PE switch (PE1), copy and paste the following commands into the switch terminal window of switch PE1:

```
set interfaces xe-0/0/12 unit 0 family inet address 10.0.0.1/24
set interfaces xe-0/0/12 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set routing-options router-id 192.168.0.1
set routing-options autonomous-system 65550
set policy-options policy-statement pplb then load-balance per-packet
set routing-options forwarding-table export pplb
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface em0.0 disable
set protocols mpls interface all
set protocols ldp interface xe-0/0/12.0
set protocols ldp interface lo0.0
```

Step-by-Step Procedure

To configure the ingress PE switch (PE1):

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@switchPE1# set xe-0/0/12 unit 0 family inet address 10.0.0.1/24
   user@switchPE1# set xe-0/0/12 unit 0 family mpls
   user@switchPE1# set lo0 unit 0 family inet address 192.168.0.1/32
   ```

2. Configure the router ID and autonomous system (AS) number.

   NOTE: We recommend that you explicitly configure the router identifier under the **[edit routing-options]** hierarchy level to prevent unpredictable behavior if the interface address on a loopback interface changes.

   ```
   [edit routing-options]
   user@switchPE1# set router-id 192.168.0.1/32
   user@switchPE1# set autonomous-system 65550
   ```

3. Configure and apply an export routing policy to the forwarding table for per-packet load balancing.

   ```
   [edit policy-options]
   ```

```
user@switchPE1# set policy-statement pplb then load-balance per-packet
[edit routing-options]
user@switchPE1# set forwarding-table export pplb
```

4. Create an OSPF area and set the loopback address to be passive.

```
[edit protocols ospf]
user@switchPE1# set area 0.0.0.0 interface all
user@switchPE1# set area 0.0.0.0 interface lo0.0 passive
user@switchPE1# set area 0.0.0.0 interface em0.0 disable
```

5. Enable MPLS on all interfaces.

```
[edit protocols mpls]
user@switchPE1# set interface all
```

6. Configure LDP on the provider-facing and loopback interfaces.

```
[edit protocols ldp]
user@switchPE1# set interface xe-0/0/12.0
user@switchPE1# set interface lo0.0
```

Results Display the results of the PE1 switch configuration:

```
user@switchPE1# show
interfaces {
    xe-0/0/12 {
        unit 0 {
            family inet {
                address 10.0.0.1/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32;
                }
            }
        }
    }
}
routing-options {
    router-id 192.168.0.1;
    autonomous-system 65550;
    forwarding-table {
        export pplb;
    }
}
protocols {
    mpls {
        interface all;
```

```
    }
    ospf {
      area 0.0.0.0 {
        interface all;
        interface lo0.0 {
          passive;
        }
        interface em0.0 {
          disable;
        }
      }
    }
    ldp {
      interface xe-0/0/12.0
      interface lo0.0;
    }
  }
  policy-options {
    policy-statement pplb {
      then {
        load-balance per-packet;
      }
    }
  }
```

Configuring the Provider Switch

CLI Quick Configuration

To quickly configure the provider switch (P), copy and paste the following commands into the switch terminal window of the P switch:

```
set interfaces xe-0/0/12 unit 0 family inet address 10.0.0.2/24
set interfaces xe-0/0/12 unit 0 family mpls
set interfaces xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members v100
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set interfaces irb unit 0 family inet address 10.0.1.2/24
set interfaces irb unit 0 family mpls
set routing-options router-id 192.168.0.2
set routing-options autonomous-system 65550
set policy-options policy-statement pplb then load-balance per-packet
set routing-options forwarding-table export pplb
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface em0.0 disable
set protocols mpls interface all
set protocols ldp interface all
set vlans v100 vlan-id 100
set vlans v100 l3-interface irb.0
```

Step-by-Step Procedure

To configure the provider switch (P):

1. Configure the physical and loopback interfaces.

```
[edit interfaces]
user@switchP#  set xe-0/0/12 unit 0 family inet address 10.0.0.2/24
```

```
user@switchP#  set xe-0/0/12 unit 0 family mpls
user@switchP# set xe-0/0/10 unit 0 family ethernet-switching interface-mode
  trunk
user@switchP# set xe-0/0/10 unit 0 family ethernet-switching vlan members v100
user@switchP# set lo0 unit 0 family inet address 192.168.0.2/32
```

2. Configure an IRB interface.

```
[edit interfaces]
user@switchP# set irb unit 0 family inet address 10.0.1.2/24
user@switchP# set irb unit 0 family mpls
```

3. Configure the router ID and AS number.

NOTE: We recommend that you explicitly configure the router identifier under the **[edit routing-options]** hierarchy level to avoid unpredictable behavior if the interface address on a loopback interface changes.

```
[edit routing-options]
user@switchP# router-id 192.168.0.2
user@switchP# set autonomous-system 65550
```

4. Configure and apply an export routing policy to the forwarding table for per-packet load balancing.

```
[edit policy-options]
user@switchP# set policy-statement pplb then load-balance per-packet
[edit routing-options]
user@switchP# set forwarding-table export pplb
```

5. Enable OSPF and set the loopback address to passive.

```
[edit protocols ospf]
user@switchP# set area 0.0.0.0 interface all
user@switchP# set area 0.0.0.0 interface lo0.0 passive
user@switchP# set area 0.0.0.0 interface em0.0 disable
```

6. Enable MPLS on all interfaces.

```
[edit protocols mpls]
user@switchP# set interface all
```

7. Configure LDP to include all interfaces.

```
[edit protocols ldp]
user@switchP# set interface all
```

8. Create the VLAN and associate the IRB interface to it.

```
[edit vlans]
user@switchP# set v100 vlan-id 100
```

```
user@switchP# set v100 l3-interface irb.0
```

NOTE: Layer 3 interfaces on trunk ports allow the interface to transfer traffic between multiple VLANs. Within a VLAN, traffic is switched, while across VLANs, traffic is routed.

Results Display the results of the provider switch configuration:

```
user@switchP# show
interfaces {
    xe-0/0/10 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members v100;
                }
            }
        }
    }
    xe-0/0/12 {
        unit 0
        family inet {
            address 10.0.0.2/24;
        }
        family mpls;
    }
    irb {
        unit 0 {
            family inet {
                address 10.0.1.2/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.2/32;
            }
        }
    }
}

routing-options {
    router-id 192.168.0.2;
    autonomous-system 65550;
    forwarding-table {
        export pplb;
    }
}
```

```
protocols {
  mpls {
    interface all;
  }
  ospf {
    area 0.0.0.0 {
      interface all;
      interface lo0.0 {
        passive;
      }
      interface em0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
  }
}

policy-options {
  policy-statement pplb {
    then {
      load-balance per-packet;
    }
  }
}

vlans {
  v100 {
    vlan-id 100;
    l3-interface irb.0;
  }
}
```

Configuring the Remote Egress PE Switch

CLI Quick Configuration

To quickly configure the remote egress PE switch (PE2), copy and paste the following commands into the switch terminal window of PE2:

```
set interfaces xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members v100
set interfaces irb unit 0 family inet address 10.0.1.3/24
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set interfaces irb unit 0 family mpls
set routing-options router-id 192.168.0.3
set routing-options autonomous-system 65550
set policy-options policy-statement pplb then load-balance per-packet
set routing-options forwarding-table export pplb
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface em0.0 disable
set protocols mpls interface all
set protocols ldp interface all
set vlans v100 vlan-id 100
set vlans v100 l3-interface irb.0
```

Step-by-Step Procedure

To configure the remote PE switch (PE2):

1. Configure the physical and loopback interfaces.

```
[edit interfaces]
user@switchPE2# set xe-0/0/10 unit 0 family ethernet-switching interface-mode
  trunk
user@switchPE2# set xe-0/0/10 unit 0 family ethernet-switching vlan members
  v100
user@switchPE2# set lo0 unit 0 family inet address 192.168.0.3/32
```

2. Configure an IRB interface.

```
[edit interfaces]
user@switchPE2# set irb unit 0 family inet address 10.0.1.3/24
user@switchPE2# set irb unit 0 family mpls
```

3. Configure the the router ID and AS number.

```
[edit routing-options]
user@switchPE2# set router-id 192.168.0.3/32
user@switchPE2# set autonomous-system 65550
```

4. Configure and apply an export routing policy to the forwarding table for per-packet load balancing.

```
[edit policy-options]
user@switchPE2# set policy-statement pplb then load-balance per-packet
[edit routing-options]
user@switchPE2# set forwarding-table export pplb
```

5. Enable OSPF.

```
[edit protocols ospf]
user@switchPE2# set area 0.0.0.0 interface all
user@switchPE2# set area 0.0.0.0 interface lo0.0 passive
user@switchPE2# set area 0.0.0.0 interface em0.0 disable
```

6. Enable MPLS on all interfaces.

```
[edit protocols mpls]
user@switchPE2# set interface all
```

7. Configure LDP to include all interfaces.

```
[edit protocols ldp]
user@switchPE2# set interface all
```

8. Create the VLAN and associate the IRB interface to it.

```
[edit vlans]
user@switchPE2# set v100 vlan-id 100
user@switchPE2# set v100 l3-interface irb.0
```

Results Display the results of the PE2 switch configuration:

```
user@switchPE2# show
interfaces {
    xe-0/0/10 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
                vlan {
                    members v100;
                }
            }
        }
    irb {
        unit 0 {
            family inet {
                address 10.0.1.3/24;
            }
            family mpls;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.3;
            }
        }
    }
}

routing-options {
    router-id 192.168.0.3;
    autonomous-system 65550;
    forwarding-table {
        export pplb;
    }
}

protocols {
    mpls {
        interface all;
    }
    ospf {
        area 0.0.0.0 {
            interface all;
            interface lo0.0 {
                passive;
            }
            interface em0.0 {
                disable;
            }
        }
    }
    ldp {
        interface all;
    }
}
```

```
policy-options {
  policy-statement pplb {
    then {
      load-balance per-packet;
    }
  }
}
vlans {
  v100 {
    vlan-id 100;
    l3-interface irb.0;
  }
}
```

Release History Table

Release	Description
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40 and Junos OS Release 17.1R1, QFX5100 switches support integrated routing and bridging (IRB) interfaces over an MPLS core network.

Related Documentation

- Configuring IRB Interfaces on Switches on page 454
- Understanding Integrated Routing and Bridging on page 445
- Understanding Bridging and VLANs on Switches on page 84
- *Understanding Layer 3 Logical Interfaces*

Example: Configuring a Large Delay Buffer on a Security Device IRB Interface

This example shows how to configure a large delay buffer on an IRB interface to help slower interfaces avoid congestion and packet dropping when they receive large bursts of traffic.

Requirements

Before you begin, enable the large buffer feature on the IRB interface and then configure a buffer size for each queue in the CoS scheduler. See *Scheduler Buffer Size Overview*.

Overview

On devices, you can configure large delay buffers on an irb interfaces.

In this example, you configure scheduler map to associate schedulers to a defined forwarding class **be-class**, **ef-class** , **af-class**, and **nc-class** using scheduler map

large-buf-sched-map. You apply scheduler maps to irb interface, and define per-unit scheduler for the IRB interface.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from the configuration mode.

```
set class-of-service scheduler-maps large-buf-sched-map forwarding-class be-class
  scheduler be-scheduler
set class-of-service scheduler-maps large-buf-sched-map forwarding-class ef-class
  scheduler ef-scheduler
set class-of-service scheduler-maps large-buf-sched-map forwarding-class af-class
  scheduler af-scheduler
set class-of-service scheduler-maps large-buf-sched-map forwarding-class nc-class
  scheduler nc-scheduler
set class-of-service interfaces irb unit 0 scheduler-map large-buf-sched-map
set interfaces irb per-unit-scheduler
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a large delay buffer on a channelized T1 interface:

1. Configure the scheduler map to associate schedulers with defined forwarding classes.

```
[edit class-of-service]
set scheduler-maps large-buf-sched-map forwarding-class be-class scheduler
  be-scheduler
set scheduler-maps large-buf-sched-map forwarding-class ef-class scheduler
  ef-scheduler
set scheduler-maps large-buf-sched-map forwarding-class af-class scheduler
  af-scheduler
set scheduler-maps large-buf-sched-map forwarding-class nc-class scheduler
  nc-scheduler
```

2. Apply the scheduler map to the IRB interface.

```
[edit ]
user@host# set interfaces irb unit 0 scheduler-map large-buf-sched-map
```

3. Define the per-unit scheduler for the irb interface.

```
[edit ]
user@host# set interfaces irb per-unit-scheduler
```

Results From configuration mode, confirm your configuration by entering the **show class-of-service** and **show chassis** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show class-of-service
  interfaces {
  irb {
    unit 0 {
      scheduler-map large-buf-sched-map;
    }
  }
}
scheduler-maps {
  large-buf-sched-map {
    forwarding-class be-class scheduler be-scheduler;
    forwarding-class ef-class scheduler ef-scheduler;
    forwarding-class af-class scheduler af-scheduler;
    forwarding-class nc-class scheduler nc-scheduler;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Large Delay Buffers Configuration

Purpose Verify that the large delay buffers are configured properly.

Action From configuration mode, enter the **show class-of-service interface irb** command.

```
user@host> show class-of-service interface irb


Physical interface: irb, Index: 132
Maximum usable queues: 8, Queues in use: 4Code point type: dscp
Scheduler map: <default>, Index :2
Congestion-notification: Disabled
Logical interface: irb.10, Index: 73
Object                  Name                    Type                    Index
Classifier              ipprec-compatibility    ip                         13
```

Meaning The large delay buffers are configured on IRB interface as expected.

Related Documentation

- Schedulers Overview on page 752
- *Default Scheduler Settings*
- *Example: Configuring and Applying Scheduler Maps*
- *Transmission Scheduling Overview*

Configuring a Set of VLANs to Act as a Switch for a Layer 2 Trunk Port

You can configure a set of VLANs that are associated with a Layer 2 trunk port. The set of VLANs function as a switch. Packets received on a trunk interface are forwarded within a VLAN that has the same VLAN identifier. A trunk interface also provides support for IRB, which provides support for Layer 2 bridging and Layer 3 IP routing on the same interface.

To configure a Layer 2 trunk port and set of VLANs, include the following statements:

```
[edit interfaces]
interface-name {
  unit number {
    family ethernet-switching {
      interface-mode access;
      vlan-members (vlan-name | vlan-tag);
    }
  }
}
interface-name {
  native-vlan-id number;
  unit number {
    family ethernet-switching {
      interface-mode trunk;
      vlan-members (vlan-name | vlan-tag);
    }
  }
}
[edit vlans ]
vlan-name {
  vlan-id number;
  vlan-id-list [ vlan-id-numbers ];
  ....
}
```

You must configure a VLAN and VLAN identifier for each VLAN associated with the trunk interface. You can configure one or more trunk or access interfaces at the **[edit interfaces]** hierarchy level. An access interface enables you to accept packets with no VLAN identifier.

Verifying Integrated Routing and Bridging Interface Status and Statistics on EX Series Switches

Purpose Determine status information and traffic statistics for integrated routing and bridging (IRB) interfaces.

Action Display IRB interfaces and their current states:

```
user@switch> show interfaces irb terse
Interface               Admin Link Proto    Local                 Remote
irb                     up    up
irb.111                 up    up   inet     10.111.111.1/24
...
```

Display Layer 2 VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs:

```
user@switch> show vlans
Routing instance        VLAN name             Tag         Interfaces
default-switch          irb                     101
                                                          ge-0/0/18.0
default-switch          support                 111
                                                          ge-0/0/18.0
...
```

Display Ethernet switching table entries for the VLAN that is attached to the IRB interface:

```
user@switch> show ethernet-switching table
MAC flags (S -static MAC, D -dynamic MAC, L -locally learned
           SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    Name                address             flags              interface
    support             00:01:02:03:04:05   S             -    ge-0/0/18.0
...
```

Display the ingress-counting statistics of an IRB interface with either the **show interfaces irb detail** command or the **show interfaces irb extensive** command. Ingress counting is displayed as **Input bytes** and **Input packets** and egress counting is displayed as **Output bytes** and **Output packets** under **Transit Statistics**.

```
user@switch> show interfaces irb.111 detail

Logical interface irb.111 (Index 65) (SNMP ifIndex 503) (HW Token 100) (Generation
 131)
Flags: SNMP-Traps 0x4000  Encapsulation: ENET2
Bandwidth: 1000mbps
Routing Instance: default-switch Bridging Domain: irb+111
Traffic statistics:
 Input bytes:    17516756
 Output bytes:   411764
 Input packets:  271745
 Output packets: 8256
Local statistics:
 Input bytes:    3240
 Output bytes:   411764
 Input packets:  54
 Output packets: 8256
Transit statistics:
 Input bytes:    17513516   0 bps
 Output bytes:   0    0 bps
 Input packets:  271745  0 pps
 Output packets: 0    0 pps
Protocol inet, MTU: 1514, Generation: 148, Route table: 0
 Flags: None
 Addresses, Flags: iS-Preferred  Is-Primary
  Destination: 10.1.1/24, Local: 10.1.1.1, Broadcast: 10.1.1.255, Generation: 136
```

Meaning

- **show interfaces irb terse** displays a list of interfaces, including IRB interfaces, and their current states (up, down).
- **show vlans** displays a list of VLANs, including any tags assigned to the VLANs and the interfaces associated with the VLANs.
- **show ethernet-switching table** displays the Ethernet switching table entries, including VLANs attached to the IRB interface.
- **show interfaces irb detail** displays IRB interface ingress counting as **Input Bytes** and **Input Packets** under **Transit Statistics**.

Related Documentation

- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) on page 456

CHAPTER 19

Configuring Virtual Routing Interfaces

Understanding Virtual Routing Instances on EX Series Switches

Virtual routing instances allow administrators to divide a Juniper Networks EX Series Ethernet Switch into multiple independent virtual routers, each with its own routing table. Splitting a device into many virtual routing instances isolates traffic traveling across the network without requiring multiple devices to segment the network.

You can use virtual routing instances to isolate customer traffic on your network and to bind customer-specific instances to customer-owned interfaces.

Virtual routing and forwarding (VRF) is often used in conjunction with Layer 3 subinterfaces, allowing traffic on a single physical interface to be differentiated and associated with multiple virtual routers. Each logical Layer 3 subinterface can belong to only one routing instance.

EX Series switches support IPv4 and IPv6 unicast and multicast VRF traffic. See Feature Explorer for details on VRF support by switch per Junos OS release.

Related Documentation

- *Understanding Layer 3 Subinterfaces*
- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 489
- Configuring Virtual Routing Instances on EX Series Switches (CLI Procedure) on page 488

Configuring Virtual Routing Instances on EX Series Switches (CLI Procedure)

Use virtual routing and forwarding (VRF) to divide an EX Series switch into multiple virtual routing instances. VRF allows you to isolate traffic traversing the network without using multiple devices to segment your network. VRF is supported on all Layer 3 interfaces.

Before you begin, make sure to set up your VLANs. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98, "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102, or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

To configure virtual routing instances:

1. Create a routing instance:

   [edit routing-instances]`user@switch#` **set** ***routing-instance-name*** instance-type **virtual-router**

   NOTE: EX Series switches only support the virtual-router instance type.

2. Bind each routing instance to the corresponding physical interfaces:

   [edit routing-instances]`user@switch#` **set** ***routing-instance-name*** **interface** ***interface-name.logical-unit-number***

3. Create the logical interfaces that are bound to the routing instance.

   - To create a logical interface with an IPv4 address:

     [edit interfaces]`user@switch#` **set** ***interface-name*** **unit** ***logical-unit-number*** **family inet address** ***ip-address***

   - To create a logical interface with an IPv6 address:

     [edit interfaces]`user@switch#` **set** ***interface-name*** **unit** ***logical-unit-number*** **family inet6 address** ***ipv6–address***

   NOTE: Do not create a logical interface using the family ethernet-switching option in this step. Binding an interface using the family ethernet-switching option to a routing instance can cause the interface to shutdown.

4. Enable VLAN tagging on each physical interface that was bound to the routing instance:

   [edit interfaces]`user@switch#` **set** ***interface-name*** vlan-tagging

Related Documentation

- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 489
- Verifying That Virtual Routing Instances Are Working on EX Series Switches on page 492
- Understanding Virtual Routing Instances on EX Series Switches on page 487

Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches

Virtual routing instances allow each EX Series switch to have multiple routing tables on a device. With virtual routing instances, you can segment your network to isolate traffic without setting up additional devices.

This example describes how to create virtual routing instances:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 9.2 or later for EX Series switches

Before you create the virtual routing instances, make sure you have:

- Configured the necessary VLANs. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98, "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102, or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

Overview and Topology

In a large office, you may need multiple VLANs to properly manage your traffic. This configuration example shows a simple topology wherein a LAN is segmented into two VLANs, each of which is associated with an interface and a virtual routing instance, on the EX Series switch. This example also shows how to use policy statements to import routes from one of the virtual routing instances to the other.

Configuration

CLI Quick Configuration

To quickly create and configure virtual routing instances, copy the following commands and paste them into the switch terminal window:

```
[edit]
set interfaces ge-0/0/3 vlan-tagging
set interfaces ge-0/0/3 unit 0 vlan-id 1030 family inet address 10.1.1.1/24
```

```
set interfaces ge-0/0/3 unit 1 vlan-id 1031 family inet address 10.1.1.1/24
set interfaces ge-0/0/1 unit 0 family inet address 10.11.1.1/24
set interfaces ge-0/0/2 unit 0 family inet address 10.12.1.1/24
set routing-instances r1 instance-type virtual-router
set routing-instances r1 interface ge-0/0/1.0
set routing-instances r1 interface ge-0/0/3.0
set routing-instances r1 routing-options instance-import import-from-r2
set routing-instances r2 instance-type virtual-router
set routing-instances r2 interface ge-0/0/2.0
set routing-instances r2 interface ge-0/0/3.1
set routing-instances r2 routing-options instance-import import-from-r1
set policy-options policy-statement import-from-r1 term 1 from instance r1
set policy-options policy-statement import-from-r1 term 1 then accept
set policy-options policy-statement import-from-r2 term 1 from instance r2
set policy-options policy-statement import-from-r2 term 1 then accept
```

Step-by-Step Procedure

To configure virtual routing instances:

1. Create a VLAN-tagged interface:

   ```
   [edit]user@switch# set interfaces ge-0/0/3 vlan-tagging
   ```

2. Create one or more subinterfaces on the interfaces to be included in each routing instance:

   ```
   [edit]user@switch# set interfaces ge-0/0/3 unit 0 vlan-id 1030 family inet address
   10.1.1.1/24
   user@switch# set interfaces ge-0/0/3 unit 1 vlan-id 1031 family inet address 10.1.1.1/24
   user@switch# set interfaces ge-0/0/1 unit 0 family inet address 10.11.1.1/24
   user@switch# set interfaces ge-0/0/2 unit 0 family inet address 10.12.1.1/24
   ```

3. Create two virtual routing instances:

   ```
   [edit]user@switch# set routing-instances r1 instance-type virtual-router
   user@switch# set routing-instances r2 instance-type virtual-router
   ```

4. Set the interfaces for the virtual routing instances:

   ```
   [edit]user@switch# set routing-instances r1 interface ge-0/0/1.0
   user@switch# set routing-instances r1 interface ge-0/0/3.0
   user@switch# set routing-instances r2 interface ge-0/0/2.0
   user@switch# set routing-instances r2 interface ge-0/0/3.1
   ```

5. Apply a policy to routes being imported into each of the virtual routing instances:

   ```
   [edit]user@switch# set routing-instances r1 routing-options instance-import
   import-from-r2
   user@switch# set routing-instances r2 routing-options instance-import import-from-r1
   ```

6. Create a policy that imports routes from routing instances r1 to r2 and another policy that imports routes from routing instances r2 to r1:

   ```
   [edit]user@switch# set policy-options policy-statement import-from-r1 term 1 from
   instance r1
   user@switch# set policy-options policy-statement import-from-r1 term 1 then accept
   ```

```
user@switch# set policy-options policy-statement import-from-r2 term 1 from instance
r2
user@switch# set policy-options policy-statement import-from-r2 term 1 then accept
```

Results Check the results of the configuration:

```
user@switch> show configuration
interfaces {
            ge-0/0/1 {
                unit 0 {
                    family inet {
                        address 10.11.1.1/24;
                    }
                }
            }
            ge-0/0/2 {
                unit 0 {
                    family inet {
                        address 10.12.1.1/24;
                    }
                }
            }
            ge-1/0/3 {
                vlan-tagging;
                unit 0 {
                    vlan-id 1030;
                    family inet {
                        address 10.1.1.1/24;
                    }
                }
                unit 1 {
                    vlan-id 1031;
                    family inet {
                        address 10.1.1.1/24;
                    }
                }
            }
        }
        policy-options {
            policy-statement import-from-r1 {
                term 1 {
                    from instance r1;
                    then accept;
                }
            }
            policy-statement import-from-r2 {
                term 1 {
                    from instance r2;
                    then accept;
                }
            }
        }
        routing-instances {
            r1 {
                instance-type virtual-router;
                interface ge-0/0/1.0;
                interface ge-0/0/3.0;
                routing-options {
                    instance-import import-from-r2;
```

```
                }
            }
            r2 {
                instance-type virtual-router;
                interface ge-0/0/2.0;
                interface ge-0/0/3.1;
                routing-options {
                    instance-import import-from-r1;
                }
            }
        }
    }
}
```

Verification

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

- Verifying That the Routing Instances Were Created on page 492

Verifying That the Routing Instances Were Created

Purpose Verify that the virtual routing instances were properly created on the switch.

Action Use the **show route instance** command:

```
user@switch> show route instance
Instance             Type
        Primary RIB                                 Active/holddown/hidden
master               forwarding
        inet.0                                      6/0/0
        iso.0                                       1/0/0
        inet6.0                                     2/0/0

...

r1                   virtual-router
        r1.inet.0                                   7/0/0

r2                   virtual-router
        r2.inet.0                                   7/0/0
```

Meaning Each routing instance created is displayed, along with its type, information about whether it is active or not, and its primary routing table.

Related Documentation

- Configuring Virtual Routing Instances on EX Series Switches (CLI Procedure) on page 488

Verifying That Virtual Routing Instances Are Working on EX Series Switches

Purpose After creating a virtual routing instance, make sure it is set up properly.

Action

1. Use the **show route instance** command to list all of the routing instances and their properties:

```
user@switch> show route instance
Instance             Type
        Primary RIB                                      Active/holddown/hidden
master               forwarding
        inet.0                                             3/0/0

__juniper_private1__ forwarding
        __juniper_private1__.inet.0                        1/0/3

__juniper_private2__ forwarding

instance1            forwarding

r1                   virtual-router
        r1.inet.0                                          1/0/0

r2                   virtual-router
        r2.inet.0                                          1/0/0
```

2. Use the **show route forwarding-table** command to view the forwarding table information for each routing instance:

```
user@switch> show route forwarding-table
Routing table: r1.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   539     2
0.0.0.0/32         perm     0                    dscd   537     1
10.1.1.0/24        ifdn     0                    rslv   579     1 ge-0/0/3.0
10.1.1.0/32        iddn     0 10.1.1.0           recv   577     1 ge-0/0/3.0
10.1.1.1/32        user     0                    rjct   539     2
10.1.1.1/32        intf     0 10.1.1.1           locl   578     2
10.1.1.1/32        iddn     0 10.1.1.1           locl   578     2
10.1.1.255/32      iddn     0 10.1.1.255         bcst   576     1 ge-0/0/3.0
233.252.0.1/32       perm     0 233.252.0.1          mcst   534     1
255.255.255.255/32 perm     0                    bcst   535     1
```

Meaning

The output confirms that the virtual routing instances are created and the links are up and displays the routing table information.

Related Documentation

- Configuring Virtual Routing Instances on EX Series Switches (CLI Procedure) on page 488
- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 489

CHAPTER 20

Configuring Multiple VLAN Registration Protocol (MVRP)

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. Use MVRP on Juniper Networks switches to dynamically register and unregister active VLANs on trunk interfaces. Using MVRP means that you do not have to manually register VLANs on all connections—that is, you do not need to explicitly bind a VLAN to each trunk interface. With MVRP, you configure a VLAN on one switch interface and the VLAN configuration is distributed through all active switches in the domain.

NOTE: 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.

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 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.

How MVRP Updates, Creates, and Deletes VLANs on Switches

When any MVRP-member VLAN is changed, that VLAN sends a protocol data unit (PDU) to all other MVRP-member active VLANs. The PDU informs the other VLANs which switches and interfaces currently belong to the sending VLAN. This way, all MVRP-member VLANs are always updated with the current VLAN state of all other MVRP-member VLANs. Timers dictate when PDUs can be sent and when switches receiving MVRP PDUs can update their MVRP VLAN information.

In addition to sending PDU updates, MVRP dynamically creates VLANs on member interfaces when a new VLAN is added to any one interface. This way, VLANs created on one member switch are propagated to other member switches as part of the MVRP message exchange process.

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 on Switches

MVRP is disabled by default on the switches and, when enabled, affects only trunk interfaces. Once you enable MVRP, all VLAN interfaces on the switch belong to MVRP (the default **normal** registration mode) and those interfaces accept PDU messages and send their own PDU messages. To prevent one or more interfaces from participating in MVRP, you can specifically configure an interface to **forbidden** registration mode instead of the default **normal** mode.

VLAN updating, dynamic VLAN configuration through MVRP, and VLAN pruning are all active on trunk interfaces when MVRP is enabled.

MRP Timers Control MVRP Updates

MVRP registration and updates are controlled by timers that are part of the MRP. The timers define when MVRP PDUs can be sent and when MVRP information can be updated on a switch.

The timers are set on a per-interface basis, and on EX Series switches that use Juniper Networks Junos operating system (Junos OS) with support for the Enhanced Layer 2 Software (ELS) configuration style, the timers are also set on a per-switch basis.

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, the value on the interface level takes precedence.

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 switch or VLAN and to inform the switching network that a switch or VLAN is leaving MVRP. These messages are communicated as part of the PDU sent by any switch interface to the other switches in the network.

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.

Compatibility Issues with Junos OS Releases of MVRP

Except in Junos OS Releases 11.2 and earlier, MVRP has conformed with IEEE standard 802.1ak and IEEE Draft 802.1Q regarding the inclusion of an extra byte in the protocol data units (PDUs) sent and received by MVRP. Table 81 on page 499 outlines the MVRP versions and whether or not each version includes the extra byte in the PDU. Table 81 on page 499 also labels each MVRP version with a scenario number, which is used throughout the remainder of this discussion for brevity.

Table 81: Junos OS MVRP Versions and Inclusion of Extra Byte in PDU

MVRP in Junos OS Releases 11.2 and Earlier For EX Series Switches That Do Not Support Enhanced Layer 2 Software (ELS) Configuration Style Scenario 1	MVRP in Junos OS Releases 11.3 and Later For EX Series Switches That Do Not Support ELS Scenario 2	MVRP in Junos OS Releases 13.2 and Later For EX Series Switches With Support For ELS Scenario 3
Includes extra byte in the PDU	By default, does not include extra byte in the PDU	By default, includes extra byte in the PDU

As a result of the non-conformance of Releases 11.2 and earlier and changes in the standards with regard to the extra byte, a compatibility issue exists between some of the Junos OS versions of MVRP. This issue can result in some versions of MVRP not recognizing PDUs without the extra byte.

To address this compatibility issue, the MVRP versions described in scenarios 2 and 3 include the ability to control whether or not the PDU includes the extra byte. Before using these controls, however, you must understand each scenario that applies to your environment and plan carefully so that you do not inadvertently create an additional compatibility issue between the MVRP versions in scenarios 2 and 3.

Table 82 on page 500 provides a summary of environments that include the various MVRP scenarios and whether or not a particular environment requires you to take action.

Table 82: MVRP Environments and Description of Required Actions

Environment	Action Required?	Action Description
Includes MVRP versions in scenario 1 only	No	—
Includes MVRP versions in scenario 2 only	No	—
Includes MVRP versions in scenario 3 only	No	—
Includes MVRP versions in scenarios 1 and 2. MVRP version in scenario 2 is in its default state.	Yes	On switches running MVRP version in scenario 2, use the **add-attribute-length-in-pdu** statement. For more information, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.
Includes MVRP versions in scenarios 1 and 3. MVRP version in scenario 3 is in its default state.	No	—
Includes MVRP versions in scenarios 2 and 3, and both versions are in their respective default states	Yes	Do one of the following: • On switches running MVRP version in scenario 2, use the **add-attribute-length-in-pdu** statement. For more information, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504. • On switches running MVRP version in scenario 3, use the **no-attribute-length-in-pdu** statement. For more information, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.

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 83 on page 500 summarizes the VLAN requirements for redundant server Node groups and network Node groups:

Table 83: 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

Determining Whether MVRP is Working

You can determine whether the switches in your network are running incompatible versions of MVRP by issuing the **show mvrp statistics** command. For more information on diagnosing and correcting this MVRP compatibility situation, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Understanding Bridging and VLANs on Switches on page 84
- Example: Configuring Automatic VLAN Administration on QFX Switches Using MVRP on page 516
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504

Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration

Multiple VLAN Registration Protocol (MVRP) is a Layer 2 messaging protocol that manages the addition, deletion, and renaming of active virtual LANs, thereby reducing network administrators' time spent on these tasks. Use MVRP on Juniper Networks MX Series routers, EX Series switches and SRX devices to dynamically register and unregister active VLANs on trunk interfaces. Using MVRP means that you do not have to manually register VLANs on all connections—that is, you do not need to explicitly bind a VLAN to each trunk interface. With MVRP, you configure a VLAN on one interface and the VLAN configuration is distributed through all active interfaces in the domain.

The primary purpose of MVRP is to manage dynamic VLAN registration in Layer 2 networks. In managing dynamic VLAN registration, MVRP also prunes VLAN information.

MVRP is an Layer 2 application protocol of the Multiple Registration Protocol (MRP) and is defined in the IEEE 802.1ak standard. MRP and MVRP were designed by IEEE to perform the same functions as Generic Attribute Registration Protocol (GARP) and GARP VLAN Registration Protocol (GVRP) while overcoming some GARP and GVRP limitations, in particular, limitations involving bandwidth usage and convergence time in large networks with large numbers of VLANs.

MVRP was created by IEEE as a replacement application for GVRP. MVRP and GVRP cannot be run concurrently to share VLAN information in a Layer 2 network.

This topic describes:

- How MVRP Works on page 502
- Using MVRP on page 502
- MVRP Registration Modes on page 503
- MRP Timers Control MVRP Updates on page 503

How MVRP Works

When any MVRP-member VLAN is changed, that VLAN sends a protocol data unit (PDU) to all other MVRP-member active VLANs. The PDU informs the other VLANs which devices and interfaces currently belong to the sending VLAN. This way, all MVRP-member VLANs are always updated with the current VLAN state of all other MVRP-member VLANs. Timers dictate when PDUs can be sent and when devices receiving MVRP PDUs can update their MVRP VLAN information.

The VLAN registration information sent by MVRP protocol data units (PDUs) includes the current VLANs membership—that is, which routers are members of which VLANs—and which router interfaces are in which VLAN. MVRP shares all information in the PDU with all routers participating in MVRP in the Layer 2 network.

MVRP stays synchronized using these PDUs. The routers in the network participating in MVRP receive these PDUs during state changes and update their MVRP states accordingly. MVRP timers dictate when PDUs can be sent and when routers receiving MVRP PDUs can update their MVRP information.

In addition to sending PDU updates, MVRP dynamically creates VLANs on member interfaces when a new VLAN is added to any one interface. This way, VLANs created on one member device are propagated to other member devices as part of the MVRP message exchange process.

VLAN information is distributed as part of the MVRP message exchange process and can be used to dynamically create VLANs, which are VLANs created on one switch and propagated to other routers as part of the MVRP message exchange process. Dynamic VLAN creation using MVRP is enabled by default, but can be disabled.

As part of ensuring that VLAN membership information is current, MVRP removes routers and interfaces from the VLAN information when they become unavailable. Pruning VLAN information has these benefits:

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

Using MVRP

MVRP is disabled by default on the devices and, when enabled, affects only trunk interfaces. Once you enable MVRP, all VLAN interfaces on the device belong to MVRP (the default **normal** registration mode) and those interfaces accept PDU messages and send their own PDU messages. To prevent one or more interfaces from participating in MVRP, you can specifically configure an interface to **forbidden** registration mode instead of the default **normal** mode.

VLAN updating, dynamic VLAN configuration through MVRP, and VLAN pruning are all active on trunk interfaces when MVRP is enabled.

MVRP Registration Modes

The MVRP registration mode defines whether an interface does or does not participate in MVRP.

The following MVRP registration modes are configurable:

- forbidden—The interface does not register or declare VLANS (except statically configured VLANs).
- normal—The interface accepts MVRP messages and participates in MVRP. This is the default registration mode setting.
- restricted—The interface ignores all MVRP JOIN messages received for VLANs that are not statically configured on the interface.

MRP Timers Control MVRP Updates

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

The following 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: **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.**

MVRP Uses MRP Messages to Transmit Device and VLAN States

MVRP uses MRP messages to register and declare MVRP states for a switch and to inform the Layer 2 network that a switch is leaving MVRP. These messages are communicated as part of the PDU to communicate the state of a particular switch interface on the Layer 2 network to the other switches in the network.

The following messages are communicated for MVRP:

- Empty—VLAN information is not being declared and is not registered.
- In—VLAN information is not being declared but is registered.
- JoinEmpty—VLAN information is being declared but not registered.

- JoinIn—VLAN information is being declared and is registered.
- Leave—VLAN information that was previously registered is being withdrawn.
- LeaveAll—All registrations will be de-registered. Participants that want to participate in MVRP will need to re-register.
- New—VLAN information is new and possibly not previously registered.

MVRP Limitations

The following limitations apply when configuring MVRP:

- MVRP works with Rapid Spanning Tree Protocol (RSTP) and Multiple Spanning Tree Protocol (MSTP), but not with VLAN Spanning Tree Protocol (VSTP).
- MVRP is allowed only on single tagged trunk ports.
- MVRP is not allowed if a physical interface has more than one logical interface.
- MVRP is only allowed if a logical has one trunk interface (unit 0).

Related Documentation

- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511

  Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on Security Devices on page 513
- *Controlling the Management State of a VLAN in MVRP Configurations*
- Verifying That MVRP Is Working Correctly on page 551
- Deploying MVRP Learning Byte at https://www.youtube.com/watch?v=C-JkzYbGPBk

Configuring Multiple VLAN Registration Protocol (MVRP) on Switches

Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a LAN. You can use MVRP on QFX switches and on EX Series switches that support or do not support ELS.

MVRP is disabled by default.

To enable MVRP or set MVRP options, follow these instructions:

- Enabling MVRP on Switches With ELS Support on page 505
- Enabling MVRP on Switches Without ELS Support on page 505
- Enabling MVRP on Switches With QFX Support on page 505
- Disabling MVRP on page 506
- Disabling Dynamic VLANs on EX Series Switches on page 506
- Configuring Timer Values on page 507

Enabling MVRP on Switches With ELS Support

This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. MVRP can only be enabled on trunk interfaces.

NOTE: For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

MVRP can only be enabled on trunk interfaces. To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name
```

Enabling MVRP on Switches Without ELS Support

This example uses Junos OS for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a LAN. You can use MVRP on EX Series switches.

MVRP is disabled by default on EX Series switches.

MVRP can only be enabled on trunk interfaces. To enable MVRP on a trunk interface:

```
[edit protocols mvrp]
user@switch# set interface all
```

To enable MVRP on a specific trunk interface:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0
```

Enabling MVRP on Switches With QFX Support

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:

This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. MVRP can only be enabled on trunk interfaces.

NOTE: For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

MVRP can only be enabled 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. Perform this procedure only if you have previously enabled MVRP.

You can disable MVRP globally only. To disable MVRP on all trunk interfaces on a switch with ELS support, use one of the following commands:

```
user@switch# deactivate protocols mvrp
user@switch# delete protocols mvrp
```

To disable MVRP on all trunk interfaces of a QFX switch, an EX switch without ELS Support or an entire QFabric system:

```
[edit protocols mvrp]
user@switch# set disable
```

To disable MVRP on a specific trunk QFX switch or an EX switch without interface support:

```
[edit protocols mvrp]
user@qfabric# set disable interface interface-name
```

```
[edit protocols mvrp]
user@switch# set disable interface xe-0/0/1.0
```

See Also

- disable on page 912
- add-attribute-length-in-pdu on page 887
- no-attribute-length-in-pdu on page 1049

Disabling Dynamic VLANs on EX Series Switches

By default, dynamic VLANs can be created on interfaces participating in MVRP. Dynamic VLANs are VLANs created on one switch that are propagated to other switches dynamically, in this case, using MVRP.

Dynamic VLAN creation through MVRP cannot be disabled per switch interface. To disable dynamic VLAN creation for interfaces participating in MVRP, you must disable it for all interfaces on the switch.

To disable dynamic VLAN creation:

```
[edit protocols mvrp]
user@switch# set no-dynamic-vlan
```

See Also
- no-dynamic-vlan on page 1050
- add-attribute-length-in-pdu on page 887
- no-attribute-length-in-pdu on page 1049

Configuring Timer Values

The timers in MVRP define the amount of time all interfaces on a switch or a specific interface wait 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 10 seconds 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.

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.

To set the join timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set join-timer milliseconds
```

```
[edit protocols mvrp]
user@switch# set interface all join-timer 300
```

To set the join timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name join-timer milliseconds
```

```
[edit protocols mvrp]
user@qfabric# set interface interface-name 300
```

```
[edit protocols mvrp]
```

```
user@switch# set interface xe-0/0/1.0 300
```

To set the leave timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set leave-timer milliseconds
```

```
[edit protocols mvrp]
user@switch# set interface all leave-timer 1200
```

To set the leave timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name leave-timer milliseconds
```

```
[edit protocols mvrp]
user@qfabric# set interface interface-name leave-timer 1200
```

To set the leaveall timer for all interfaces on the switch:

```
[edit protocols mvrp]
user@switch# set leaveall-timer seconds
```

```
[edit protocols mvrp]
user@qfabric# set interface interface-name leaveall-timer 12000
```

```
[edit protocols mvrp]
user@switch# set interface all leaveall-timer 12000
```

To set the leaveall timer for a specific interface:

```
[edit protocols mvrp]
user@switch# set interface interface-name leaveall-timer seconds
```

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 leaveall-timer 12000
```

See Also

- join-timer (MVRP) on page 1010
- leave-timer (MVRP) on page 1020
- leaveall-timer (MVRP) on page 1022

Configuring Passive Mode on QFX Switches

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
```

Configuring MVRP Registration Mode on EX Switches

NOTE: Not supported in QFabric.

The default MVRP registration mode for any interface participating in MVRP is normal. An interface in normal registration mode participates in MVRP when MVRP is enabled on the switch.

You can change the registration mode of a specific interface to **forbidden**. An interface in forbidden registration mode does not participate in MVRP even if MVRP is enabled on the switch.

To set an interface to forbidden registration mode:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration forbidden
```

```
[edit protocols mvrp]
user@switch# set interface all registration forbidden
```

To set an interface to normal registration mode:

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration normal
```

```
[edit protocols mvrp]
user@switch# set interface xe-0/0/1.0 registration normal
```

To set all interfaces to normal registration mode:

```
[edit protocols mvrp]
user@switch# set interface all registration normal
```

See Also

- registration on page 1095

Using MVRP in a Mixed-Release EX Series Switching Network

Except in Junos OS Releases 11.2 and earlier, MVRP has conformed with IEEE standard 802.1ak and IEEE Draft 802.1Q regarding the inclusion of an extra byte in the protocol data units (PDUs) sent and received by MVRP.

As a result of the non-comformance of releases 11.2 and earlier and changes in the standards regarding the extra byte, the following mixed environments can arise in EX Series switches without ELS support:

- Mixed environment A: MVRP in Junos OS Releases 11.2 and earlier includes the extra byte, while MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style does not include the extra byte.
- Mixed environment B: MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for ELS includes the extra byte, while MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support ELS does not include the extra byte.

As a result of changes in the standards with regard to the extra byte, MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for the Enhanced Layer 2 Software (ELS) includes the extra byte, while MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support ELS does not include the extra byte. A

compatibility issue arises, wherein the ELS version of MVRP does not recognize PDUs without the extra byte sent by the non-ELS version of MVRP.

A compatibility issue arises in mixed environments A and B, wherein the versions of MVRP that include the extra byte do not recognize PDUs that do not include the extra byte.

If your network has a mix of MVRP versions, you can alter MVRP on the switches running Release 11.3 and later on switches that do not support ELS so they include the extra byte in the PDU and are therefore, compatible with the other MVRP versions.

A compatibility issue arises in mixed environments A and B, wherein the versions of MVRP that include the extra byte do not recognize PDUs that do not include the extra byte.

For more information about these issues, see "Understanding Multiple VLAN Registration Protocol (MVRP)" on page 496.

To make MVRP on on switches that do not support ELS (Release 11.3 or later) compatible with MVRP in the other releases:

```
[edit protocols mvrp]
user@switch# set add-attribute-length-in-pdu
```

If your network includes a mix of EX Series switches running ELS and non-ELS versions of MVRP, you can eliminate the compatibility issue by entering the following command on the switches running the ELS version of MVRP:

```
[edit protocols mvrp]
user@switch# set no-attribute-length-in-pdu
```

The **no-attribute-length-in-pdu** statement prevents the ELS version of MVRP from sending PDUs with the extra byte, thereby eliminating the compatibility issue with the non-ELS version of MVRP.

You can recognize an MVRP version compatibility issue by observing the switch running the ELS version of MVRP. Because a switch running the ELS version of MVRP cannot interpret an unmodified PDU from a switch running the non-ELS version of MVRP, the switch will not add VLANs from the non-ELS version of MVRP. When you use the **show mvrp statistics** command in the ELS version of MVRP, the values for **Received Join Empty** and **Received Join In** will incorrectly display zero, even though the value for the **Received MVRP PDUs without error** has been increased. Another indication that MVRP is having a version compatibility issue is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the ELS version of MVRP.

Related Documentation

- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496
- Example: Configuring Automatic VLAN Administration on QFX Switches Using MVRP on page 516
- Verifying That MVRP Is Working Correctly on Switches on page 548
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support on page 521

- Verifying That MVRP Is Working Correctly on EX Series Switches with ELS Support on page 550
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Verifying That MVRP Is Working Correctly on Switches on page 548

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration

Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a Layer 2 network. You can use MVRP on MX Series routers or on EX Series switches.

MVRP is disabled by default on MX Series routers and EX Series switches.

To enable MVRP or set MVRP options, follow these instructions:

- Enabling MVRP on page 511
- Disabling MVRP on page 511
- Changing the Registration Mode to Disable Dynamic VLANs on page 511
- Configuring Timer Values on page 512
- Configuring the Multicast MAC Address for MVRP on page 513
- Configuring an MVRP Interface as a Point-to-Point Interface on page 513
- Configuring MVRP Tracing Options on page 513

Enabling MVRP

MVRP can only be enabled on trunk interfaces.

To enable MVRP on a specific trunk interface (here, interface **ge-3/0/5**):

```
[edit protocols mvrp]
user@host# set interface ge-3/0/5
```

Disabling MVRP

MVRP is disabled by default. You only need to perform this procedure if you have previously enabled MVRP.

To disable MVRP on all trunk interfaces, use one of the following:

```
[edit]
user@host# deactivate protocols mvrp
user@host# delete protocols mvrp
```

Changing the Registration Mode to Disable Dynamic VLANs

When the registration mode for an interface is set to **normal** (the default), dynamic VLANs are created on interfaces participating in MVRP. The dynamic VLANs created on one

router or switch are then propagated by means of MVRP to other routers or switches in a topology.

However, dynamic VLAN creation through MVRP can be disabled for all trunk interfaces or for individual trunk interfaces.

For information about disabling dynamic VLAN creation on an interface so that the interface does not register and does not participate in MVRP, see *Controlling the Management State of a VLAN in MVRP Configurations* .

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 router or switch after receiving an MVRP PDU:

- The join timer controls the amount of time the router or switch waits to accept a registration request.
- The leave timer controls the period of time that the router or switch waits in the Leave state before changing to the unregistered state.
- 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 a specific interface:

```
[edit protocols mvrp]
user@host# set interface ge-3/0/5 join-timer 300
```

To set the leave timer for a specific interface:

```
[edit protocols mvrp]
user@host# set interface ge-3/0/5 leave-timer 1200
```

To set the leaveall timer for a specific interface:

```
[edit protocols mvrp]
user@host# set interface ge-3/0/5 leaveall-timer 12000
```

See Also

- join-timer (MVRP) on page 1010
- leave-timer (MVRP) on page 1020
- leaveall-timer (MVRP) on page 1022

Configuring the Multicast MAC Address for MVRP

MVRP uses the customer MVRP multicast MAC address when MVRP is enabled. However, you can configure MVRP to instead use the provider MVRP multicast MAC address.

To configure MVRP to use the provider MVRP multicast MAC address:

```
[edit protocols mvrp]
user@host# set bpdu-destination-mac-address provider-bridge-group;
```

See Also
- bpdu-destination-mac-address on page 895

Configuring an MVRP Interface as a Point-to-Point Interface

Specify that a configured interface is connected point-to-point. If specified, a point-to-point subset of the MRP state machine provides a simpler and more efficient method to accelerate convergence on the network.

To specify that an MVRP interface is point-to-point (here, interface **ge-3/0/5**):

```
[edit protocols mvrp]
user@host# set interface ge-3/0/5 point-to-point;
```

See Also
- point-to-point (MVRP) on page 1065

Configuring MVRP Tracing Options

Set MVRP protocol-level tracing options.

To specify MVRP protocol tracing (here, the file is **/var/log/mvrp-log**, size is **2m**, number of files is **28**, the option **world-readable** indicates the log can be read by user, and MVRP is flagging **events**):

```
[edit protocols mvrp]
user@host# edit traceoptions file /var/log/mvrp-log size 2m files 28 world-readable flag
  events
```

See Also
- traceoptions (MVRP) on page 1123

Related Documentation
- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*

Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on Security Devices

Starting in Junos OS Release 15.1X49-D80, Multiple VLAN Registration Protocol (MVRP) to manage dynamic VLAN registration is supported on SRX1500 devices. Multiple VLAN Registration Protocol (MVRP) is used to manage dynamic VLAN registration in a Layer 2 network. You can configure MVRP on SRX Series devices.

MVRP is disabled by default on SRX Series devices.

To enable MVRP and to set MVRP options, follow these instructions:

Enabling MVRP

MVRP can be enabled only on trunk interfaces.

To enable MVRP on a specific trunk interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1
```

Changing the Registration Mode to Disable Dynamic VLANs

When the registration mode for an interface is set to **normal** (the default), dynamic VLANs are created on interfaces participating in MVRP. The dynamic VLANs created on one SRX Series device are then propagated by means of MVRP to other SRX Series devices in the topology.

However, dynamic VLAN creation through MVRP can be disabled for all trunk interfaces or for individual trunk interfaces.

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 router or switch after receiving an MVRP PDU:

- The join timer controls the amount of time the router or switch waits to accept a registration request.
- The leave timer controls the period of time that the router or switch waits in the Leave state before changing to the unregistered state.
- 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 60 seconds 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 at 300 ms for a specific interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 join-timer (MVRP) 300
```

To set the leave timer at 400 ms for a specific interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 leave-timer 400
```

To set the leaveall timer at 20 seconds for a specific interface (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 leaveall-timer 20
```

See Also
- join-timer (MVRP) on page 1010
- leave-timer (MVRP) on page 1020
- leaveall-timer (MVRP) on page 1022

Configuring the Multicast MAC Address for MVRP

MVRP uses the customer MVRP multicast MAC address when MVRP is enabled. However, you can configure MVRP to use the provider MVRP multicast MAC address instead.

To configure MVRP to use the provider MVRP multicast MAC address:

```
[edit protocols mvrp]
user@host# set bpdu-destination-mac-address provider-bridge-group;
```

See Also
- bpdu-destination-mac-address on page 895

Configuring an MVRP Interface as a Point-to-Point Interface

Specify that a configured interface is connected point-to-point. If specified, a point-to-point subset of the MRP state machine provides a simpler and more efficient method to accelerate convergence on the network.

To specify that an MVRP interface is point-to-point (here, interface ge-0/0/1):

```
[edit protocols mvrp]
user@host# set interface ge-0/0/1 point-to-point (MVRP);
```

See Also
- point-to-point (MVRP) on page 1065

Configuring MVRP Tracing Options

Set MVRP protocol-level tracing options.

To specify MVRP protocol tracing (here, the file is **/var/log/mvrp-log**, size is **2m**, number of files is **28**, the option **world-readable** indicates the log can be read by user, and MVRP is flagging **events**):

```
[edit protocols mvrp]
user@host# edit traceoptions file /var/log/mvrp-log size 2m files 28 world-readable flag
  events
```

Disabling MVRP

MVRP is disabled by default. You need to perform this procedure only if MVRP is previously enabled.

To disable MVRP on all trunk interfaces, use one of the following commands:

```
[edit]
user@host# deactivate protocols mvrp
user@host# delete protocols mvrp
```

See Also

- Understanding VLANs on page 769

Example: Configuring Automatic VLAN Administration on QFX Switches 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 84 on page 517 explains the components of the example topology.

Table 84: 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 521

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 496

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support

NOTE: This example uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches" on page 535. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

As a network expands and the number of clients and VLANs increases, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple

EX Series switches becomes increasingly difficult. However, you can automate VLAN administration by enabling Multiple VLAN Registration Protocol (MVRP) on the network.

MVRP also dynamically creates VLANs, further simplifying the network overhead required to statically configure VLANs.

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

This example describes how to use MVRP to automate administration of VLAN membership changes within your network and how to use MVRP to dynamically create VLANs:

Requirements

This example uses the following hardware and software components:

- Two EX Series access switches
- One EX Series distribution switch
- Junos OS Release 13.2X50-D10 or later for EX Series switches

Before you configure MVRP on an interface, you must enable one of the following spanning tree protocols on that interface:

- Rapid Spanning-Tree Protocol (RSTP). For more information about RSTP, see *Understanding RSTP* .
- Multiple Spanning-Tree Protocol (MSTP). For more information about MSTP, see *Understanding MSTP*.

Overview and Topology

MVRP is used to manage dynamic VLAN registration in a LAN. It can also be used to dynamically create VLANs.

This example uses MVRP to dynamically create VLANs on the switching network. Alternatively, you can disable dynamic VLAN creation and create VLANs statically. Enabling MVRP on the trunk interface of each switch in your switching network ensures that the active VLAN information for the switches in the network is propagated to each switch through the trunk interfaces, assuming dynamic VLAN creation is enabled for MVRP.

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 are configured on the switch but 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.

NOTE: This example shows a network with three VLANs: **finance**, **sales**, and **lab**. All three VLANs are running the same version of Junos OS. If switches in this network were running a mix of Junos OS releases that included Release 11.3, additional configuration would be necessary—see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504 for details.

Figure 33 on page 524 shows MVRP configured on two access switches and one distribution switch.

Figure 33: MVRP Configured on Two Access Switches and One Distribution Switch for Automatic VLAN Administration

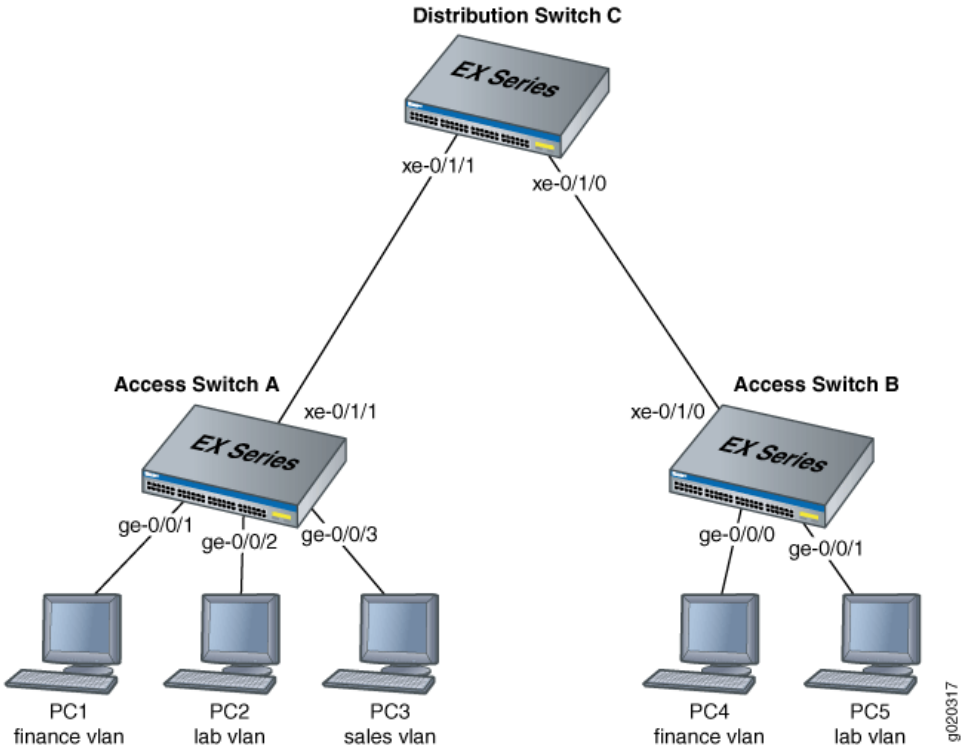


Table 84 on page 517 explains the components of the example topology.

Table 85: Components of the Network Topology

Settings	Settings
Switch hardware	• Access Switch A • Access Switch B • Distribution Switch C
VLAN names and tag IDs	**finance**, tag **100** **lab**, tag **200** **sales**, tag **300**

Table 85: Components of the Network Topology (continued)

Settings	Settings
Interfaces	Access Switch A interfaces: • ge-0/0/1—Connects PC1 to access Switch A. • ge-0/0/2—Connects PC2 to access Switch A. • ge-0/0/3—Connects PC3 to access Switch A. • xe-0/1/1—Connects access Switch A to distribution Switch C (trunk). Access Switch B interfaces: • ge-0/0/0—Connects PC4 to access Switch B. • ge-0/0/1—Connects PC5 to access Switch B. • ge-0/0/2—Reserved for future use, • xe-0/1/0—Connects access Switch B to distribution Switch C. (trunk) Distribution Switch C interfaces: • xe-0/1/1—Connects distribution Switch C to access Switch A. (trunk) • xe-0/1/0—Connects distribution Switch C to access Switch B. (trunk)

Configuring VLANs and MVRP on Access Switch A

To configure VLANs on the switch, bind access interfaces to the VLANs, and enable MVRP on the trunk interface of access Switch A, perform these tasks:

CLI Quick Configuration

To quickly configure access Switch A for MVRP, copy the following commands and paste them into the switch terminal window of Switch A:

```
[edit]
set vlans finance vlan-id 100
set vlans lab vlan-id 200
set vlans sales vlan-id 300
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members finance
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lab
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/1/1 unit 0 family ethernet-switching interface-mode trunk
set protocols mvrp interface xe-0/1/1
```

NOTE: This example uses default MVRP timers. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms (10 seconds) for the leaveall timer. We recommend retaining the use of default timer values as modifying timers to inappropriate values might cause an imbalance in the operation of MVRP. However, if you choose to change the default settings, 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.

Step-by-Step Procedure

To configure access Switch A for MVRP:

1. Configure the finance VLAN:

```
[edit]
user@Access-Switch-A# set vlans finance vlan-id 100
```

2. Configure the lab VLAN:

```
[edit]
user@Access-Switch-A# set vlans lab vlan–id 200
```

3. Configure the sales VLAN:

```
[edit]
user@Access-Switch-A# set vlans sales vlan–id 300
```

4. Configure an Ethernet interface as a member of the finance VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan
members finance
```

5. Configure an Ethernet interface as a member of the lab VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan
members lab
```

6. Configure an Ethernet interface as a member of the sales VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan
members sales
```

7. Configure a trunk interface:

```
[edit]
user@Access-Switch-A# set interfaces xe-0/1/1 unit 0 family ethernet-switching
interface-mode trunk
```

8. Enable MVRP on the trunk interface:

```
[edit]
user@Access-Switch-A# set protocols mvrp interface xe-0/1/1
```

Results

Check the results of the configuration on Switch A:

```
[edit]
user@Access-Switch-A# show
interfaces {
  ge-0/0/1 {
    unit 0 {
      family ethernet-switching {
        vlan {
```

```
            members finance;
          }
        }
      }
    }
    ge-0/0/2 {
      unit 0 {
        family ethernet-switching {
          vlan {
            members lab;
          }
        }
      }
    }
    ge-0/0/3 {
      unit 0 {
        family ethernet-switching {
          vlan {
            members sales;
          }
        }
      }
    }
    xe-0/1/1 {
      unit 0 {
        family ethernet-switching {
          interface-mode trunk;
        }
      }
    }
}
protocols {
    mvrp {
      interface xe-0/1/1;
    }
}
vlans {
    finance {
      vlan-id 100;
    }
    lab {
      vlan-id 200;
    }
    sales {
      vlan-id 300;
    }
}
```

Configuring VLANs and MVRP on Access Switch B

To configure three VLANs on the switch, bind access interfaces for PC4 and PC5 to the VLANs, and enable MVRP on the trunk interface of access Switch B, perform these tasks:

CLI Quick Configuration

To quickly configure Access Switch B for MVRP, copy the following commands and paste them into the switch terminal window of Switch B:

```
[edit]
set vlans finance vlan-id 100
set vlans lab vlan-id 200
set vlans sales vlan-id 300
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members finance
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lab
set interfaces xe-0/1/0 unit 0 family ethernet-switching interface-mode trunk
set protocols mvrp interface xe-0/1/0
```

Step-by-Step Procedure

To configure access Switch B for MVRP:

1. Configure the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans finance vlan-id 100
   ```

2. Configure the lab VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans lab vlan–id 200
   ```

3. Configure the sales VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans sales vlan–id 300
   ```

4. Configure an Ethernet interface as a member of the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan
   members finance
   ```

5. Configure an Ethernet interface as a member of the lab VLAN:

   ```
   [edit]
   user@Access-Switch-B# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan
   members lab
   ```

6. Configure a trunk interface:

   ```
   user@Access-Switch-B# set interfaces xe-0/1/0 unit 0 family ethernet-switching
   interface-mode trunk
   ```

7. Enable MVRP on the trunk interface:

   ```
   [edit]
   user@Access-Switch-B# set protocols mvrp xe-0/1/0
   ```

NOTE: This example uses default MVRP timers. The default values associated with each MVRP timer are: 200 ms for the join timer, 1000 ms for the leave timer, and 10000 ms (10 seconds) for the leaveall timer. We recommend retaining the use of default timer values as 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.

Results Check the results of the configuration for Switch B:

```
[edit]
user@Access-Switch-B# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members finance;
                }
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members lab;
                }
            }
        }
    }
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                interface-mode trunk;
            }
        }
    }
}

protocols {
    mvrp {
        interface xe-0/1/0;
    }
}
vlans {
    finance {
```

```
    vlan-id 100;
  }
  lab {
    vlan-id 200;
  }
  sales {
    vlan-id 300;
  }
}
```

Configuring VLANS and MVRP on Distribution Switch C

CLI Quick Configuration

To quickly configure distribution Switch C for MVRP, copy the following commands and paste them into the switch terminal window of distribution Switch C:

```
[edit]
set interfaces xe-0/1/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces xe-0/1/0 unit 0 family ethernet-switching interface-mode trunk
set protocols mvrp interface xe-0/1/1
set protocols mvrp interface xe-0/1/0
```

Step-by-Step Procedure

To configure distribution Switch C for MVRP:

1. Configure the trunk interface to access Switch A:

   ```
   [edit]
   user@Distribution-Switch-C# set interfaces xe-0/1/1 unit 0 family ethernet-switching
   interface-mode trunk
   ```

2. Configure the trunk interface to access Switch B:

   ```
   [edit]
   user@Distribution-Switch-C# set interfaces xe-0/1/0 unit 0 family ethernet-switching
   interface-mode trunk
   ```

3. Enable MVRP on the trunk interface for **xe-0/1/1** :

   ```
   [edit]
   user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/1
   ```

4. Enable MVRP on the trunk interface for **xe-0/1/0** :

   ```
   [edit]
   user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/0
   ```

Results

Check the results of the configuration for Switch C:

```
[edit]
user@Distribution Switch-C# show
interfaces {
  xe-0/1/0 {
    unit 0 {
      family ethernet-switching {
```

```
            interface-mode trunk;
          }
        }
      }
      xe-0/1/1 {
        unit 0 {
          family ethernet-switching {
            interface-mode trunk;
          }
        }
      }
    }
    protocols {
      mvrp {
        interface xe-0/1/0;
        interface xe-0/1/1;
      }
```

Verification

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

- Verifying That MVRP Is Enabled on Access Switch A on page 531
- Verifying That MVRP Is Updating VLAN Membership on Access Switch A on page 532
- Verifying That MVRP Is Enabled on Access Switch B on page 532
- Verifying That MVRP Is Updating VLAN Membership on Access Switch B on page 533
- Verifying That MVRP Is Enabled on Distribution Switch C on page 534
- Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C on page 534

Verifying That MVRP Is Enabled on Access Switch A

Purpose Verify that MVRP is enabled on the switch.

Action Show the MVRP configuration:

```
user@Access-Switch-A> show mvrp
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/1         200    1000     10000
```

Meaning The results show that MVRP is enabled on the trunk interface of Switch A and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch A

Purpose Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch A.

Action List Ethernet switching interfaces on the switch:

```
user@Access-Switch-A> show ethernet-switching interface
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
ge-0/0/1.0                    65535                                 tagged
             finance    100
                               65535    Forwarding
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
ge-0/0/2.0                    65535                                 tagged
             lab        200
                               65535    Forwarding
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
ge-0/0/3.0                    65535                                   tagged
             sales      300
                               65535    Forwarding
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-0/1/1.0                    65535                                   tagged
             finance    100
                               65535    Forwarding
             lab        200
                               65535    Forwarding
```

Meaning MVRP has automatically added **finance** and **lab** as VLAN members on the trunk interface because they are being advertised by access Switch B.

Verifying That MVRP Is Enabled on Access Switch B

Purpose Verify that MVRP is enabled on the switch.

Action Show the MVRP configuration:

```
user@Access-Switch-B> show mvrp
```

```
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/0         200    1000     10000
```

Meaning The results show that MVRP is enabled on the trunk interface of Switch B and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch B

Purpose Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch B.

Action List Ethernet switching interfaces on the switch:

```
user@Access-Switch-B> show ethernet-switching interface
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
ge-0/0/0.0                    65535                                 tagged
             finance    100
                               65535    Forwarding
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
ge-0/0/1.0                    65535                                 tagged
             lab        200
                               65535    Forwarding
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-0/1/0.0                    65535                                   tagged
             finance    100
                               65535    Forwarding
             lab        200
                               65535    Forwarding
             sales      300
                               65535    Forwarding
```

Meaning MVRP has automatically added **finance**, **lab**, and **sales** as VLAN members on the trunk interface because they are being advertised by access Switch A.

Verifying That MVRP Is Enabled on Distribution Switch C

Purpose Verify that MVRP is enabled on the switch.

Action Show the MVRP configuration:

```
user@Distribution-Switch-C> show mvrp
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/1         200    1000     10000
  xe-0/1/0         200    1000     10000
```

Meaning The results show that MVRP is enabled on the trunk interfaces of Switch C and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C

Purpose Verify that MVRP is updating VLAN membership on distribution Switch C by displaying the Ethernet switching interfaces and associated VLANs on distribution Switch C.

Action List the Ethernet switching interfaces on the switch:

```
user@Distribution-Switch-C> show ethernet-switching interface
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-0/1/1.0                    65535                                 tagged
             mvrp_100
                               65535    Forwarding
             mvrp_200
                               65535    Forwarding
             mvrp_300
                               65535    Forwarding
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-0/1/0.0                    65535                                   tagged
             mvrp_100
                               65535    Forwarding
             mvrp_200
                               65535    Forwarding
```

List the VLANs that were created dynamically using MVRP on the switch:

```
user@Distribution-Switch-C> show mvrp dynamic-vlan-memberships
```

```
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN ID          Interfaces
100              xe-0/1/1.0
                 xe-0/1/0.0
200              xe-0/1/1.0
                 xe-0/1/0.0
300              xe-0/1/1.0
```

Note that this scenario does not have any fixed registration, which is typical when MVRP is enabled.

Meaning Distribution Switch C has two trunk interfaces. Interface **xe-0/1/1.0** connects Distribution Switch C to Access Switch A and is, therefore, updated to show that it is a member of all the VLANs that are active on Switch A. Any traffic for those VLANs will be passed on from Switch C to Switch A, through interface **xe-0/1/1.0**. Interface **xe-0/1/0.0** connects Switch C to Switch B and is updated to show that it is a member of the two VLANs that are active on Switch B. Thus, Switch C sends traffic for **finance** and **lab** to both Switch A and Switch B. But Switch C sends traffic for **sales** only to Switch A.

Switch C also has three dynamic VLANs created using MVRP: **mvrp_100**, **mvrp_200**, and **mvrp_300**. The dynamically created VLANs **mvrp_100** and **mvrp_200** are active on interfaces **xe-0/1/1.0** and **xe-0/1/0.0**, and dynamically created VLAN **mvrp_300** is active on interface **xe-0/1/1.0**.

Related Documentation

- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496

Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches

NOTE: This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support" on page 521. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

As a network expands and the number of clients and VLANs increases, VLAN administration becomes complex and the task of efficiently configuring VLANs on multiple EX Series switches becomes increasingly difficult. To automate VLAN administration, you can enable Multiple VLAN Registration Protocol (MVRP) on the network.

MVRP also dynamically creates VLANs, further simplifying the network overhead required to statically configure VLANs.

NOTE: Only trunk interfaces can be enabled for MVRP.

This example describes how to use MVRP to automate administration of VLAN membership changes within your network and how to use MVRP to dynamically create VLANs:

Requirements

This example uses the following hardware and software components:

- Two EX Series access switches
- One EX Series distribution switch
- Junos OS Release 10.0 or later for EX Series switches

Overview and Topology

MVRP is used to manage dynamic VLAN registration in a LAN. It can also be used to dynamically create VLANs.

This example uses MVRP to dynamically create VLANs on the switching network. You can disable dynamic VLAN creation and create VLANs statically, if desired. Enabling MVRP on the trunk interface of each switch in your switching network ensures that the active VLAN information for the switches in the network is propagated to each switch through the trunk interfaces, assuming dynamic VLAN creation is enabled for MVRP.

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.

NOTE: This example shows a network with three VLANs: **finance**, **sales**, and **lab**. All three VLANs are running the same version of Junos OS. If switches in this network were running a mix of Junos OS releases that included Release 11.3, additional configuration would be necessary—see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504 for details.

Access Switch A has been configured to support all three VLANS and all three VLANS are active, bound to interfaces that are connected to personal computers:

- **ge-0/0/1**—Connects PC1 as a member of **finance**, VLAN ID 100
- **ge-0/0/2**—Connects PC2 as a member of **lab**, VLAN ID 200
- **ge-0/0/3**—Connects PC3 as a member of **sales**, VLAN ID 300

Access Switch B has also been configured to support three VLANS. However, currently only two VLANs are active, bound to interfaces that are connected to personal computers:

- **ge-0/0/0**—Connects PC4 as a member of **finance**, VLAN ID 100
- **ge-0/0/1**—Connects PC5 as a member of **lab**, VLAN ID 200

Distribution Switch C learns the VLANs dynamically using MVRP through the connection to the access switches. Distribution Switch C has two trunk interfaces:

- **xe-0/1/1**—Connects the switch to access Switch A.
- **xe-0/1/0**—Connects the switch to access Switch B.

Figure 33 on page 524 shows MVRP configured on two access switches and one distribution switch.

Figure 34: MVRP Configured on Two Access Switches and One Distribution Switch for Automatic VLAN Administration

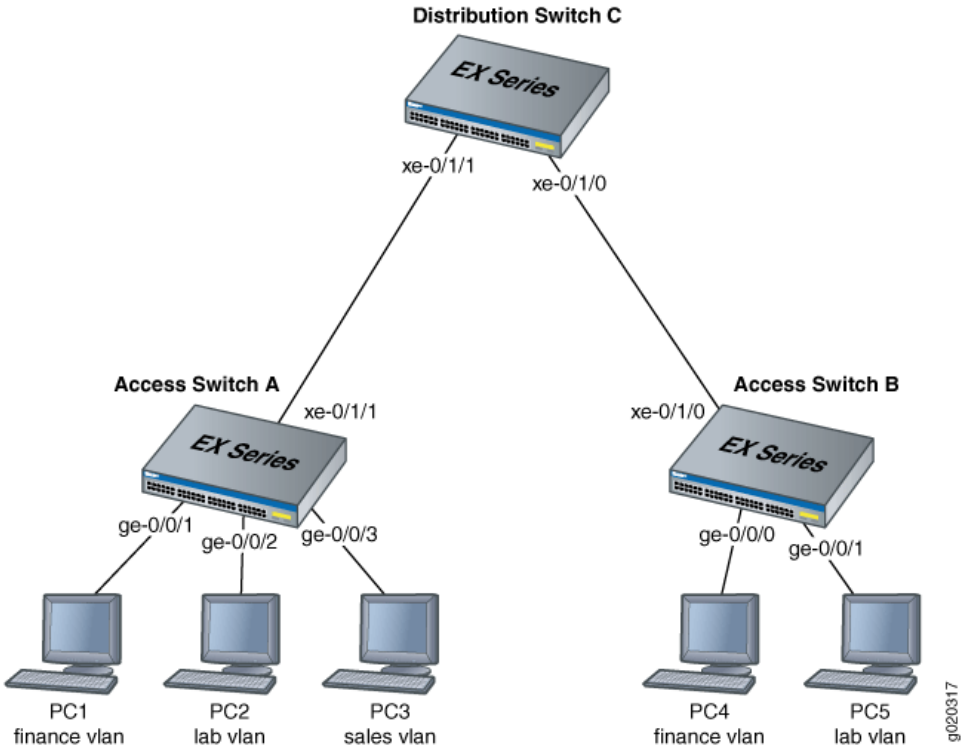


Table 84 on page 517 explains the components of the example topology.

Table 86: Components of the Network Topology

Settings	Settings
Switch hardware	• Access Switch A • Access Switch B • Distribution Switch C
VLAN names and tag IDs	**finance**, tag **100** **lab**, tag **200** **sales**, tag **300**

Table 86: Components of the Network Topology (continued)

Settings	Settings
Interfaces	Access Switch A interfaces: • **ge-0/0/1**—Connects PC1 to access Switch A. • **ge-0/0/2**—Connects PC2 to access Switch A. • **ge-0/0/3**—Connects PC3 to access Switch A. • **xe-0/1/1**—Connects access Switch A to distribution Switch C (trunk). Access Switch B interfaces: • **ge-0/0/0**—Connects PC4 to access Switch B. • **ge-0/0/1**—Connects PC5 to access Switch B. • **xe-0/1/0**—Connects access Switch B to distribution Switch C. (trunk) Distribution Switch C interfaces: • **xe-0/1/1**—Connects distribution Switch C to access Switch A. (trunk) • **xe-0/1/0**—Connects distribution Switch C to access Switch B. (trunk)

Configuring VLANs and MVRP on Access Switch A

To configure VLANs on the switch, bind access interfaces to the VLANs, and enable MVRP on the trunk interface of access Switch A, perform these tasks:

CLI Quick Configuration

To quickly configure access Switch A for MVRP, copy the following commands and paste them into the switch terminal window of Switch A:

```
[edit]
set vlans finance vlan-id 100
set vlans lab vlan-id 200
set vlans sales vlan-id 300
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members finance
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members lab
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members sales
set interfaces xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
set protocols mvrp interface xe-0/1/1.0
```

NOTE: As recommended as a best practice, default MVRP timers are used in this example. 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 configure access Switch A for MVRP:

1. Configure the finance VLAN:

```
[edit]
user@Access-Switch-A# set vlans finance vlan-id 100
```

2. Configure the lab VLAN:

```
[edit]
user@Access-Switch-A# set vlans lab vlan–id 200
```

3. Configure the sales VLAN:

```
[edit]
user@Access-Switch-A# set vlans sales vlan–id 300
```

4. Configure an Ethernet interface as a member of the finance VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan
members finance
```

5. Configure an Ethernet interface as a member of the lab VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan
members lab
```

6. Configure an Ethernet interface as a member of the sales VLAN:

```
[edit]
user@Access-Switch-A# set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan
members sales
```

7. Configure a trunk interface:

```
[edit]
user@Access-Switch-A# set interfaces xe-0/1/1 unit 0 family ethernet-switching port-mode
trunk
```

8. Enable MVRP on the trunk interface:

```
[edit]
user@Access-Switch-A# set protocols mvrp interface xe-0/1/1.0
```

Results

Check the results of the configuration on Switch A:

```
[edit]
user@Access-Switch-A# show
interfaces {
  ge-0/0/1 {
    unit 0 {
      family ethernet-switching {
        vlan {
```

```
            members finance;
          }
        }
      }
    }
    ge-0/0/2 {
      unit 0 {
        family ethernet-switching {
          vlan {
            members lab;
          }
        }
      }
    }
    ge-0/0/3 {
      unit 0 {
        family ethernet-switching {
          members sales;
          }
        }
      }
    }
    xe-0/1/1 {
      unit 0 {
        family ethernet-switching {
          port-mode trunk;
        }
      }
    }
  }
  protocols {
    mvrp {
      interface xe-0/1/1.0;
    }
  }
  vlans {
    finance {
      vlan-id 100;
    }
    lab {
      vlan-id 200;
    }
    sales {
      vlan-id 300;
    }
  }
```

Configuring VLANs and MVRP on Access Switch B

To configure three VLANs on the switch, bind access interfaces for PC4 and PC5 to the VLANs, and enable MVRP on the trunk interface of access Switch B, perform these tasks:

CLI Quick Configuration

To quickly configure Access Switch B for MVRP, copy the following commands and paste them into the switch terminal window of Switch B:

```
[edit]
set vlans finance vlan-id 100
set vlans lab vlan-id 200
set vlans sales vlan-id 300
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members finance
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members lab
set interfaces xe-0/1/0 unit 0 family ethernet-switching port-mode trunk
set protocols mvrp interface xe-0/1/0.0
```

Step-by-Step Procedure

To configure access Switch B for MVRP:

1. Configure the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans finance vlan-id 100
   ```

2. Configure the lab VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans lab vlan–id 200
   ```

3. Configure the sales VLAN:

   ```
   [edit]
   user@Access-Switch-B# set vlans sales vlan–id 300
   ```

4. Configure an Ethernet interface as a member of the finance VLAN:

   ```
   [edit]
   user@Access-Switch-B# set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan
   members finance
   ```

5. Configure an Ethernet interface as a member of the lab VLAN:

   ```
   [edit]
   user@Access-Switch-B# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan
   members lab
   ```

6. Configure a trunk interface:

   ```
   user@Access-Switch-B# set interfaces xe-0/1/0 unit 0 family ethernet-switching port-mode
   trunk
   ```

7. Enable MVRP on the trunk interface:

   ```
   [edit]
   user@Access-Switch-B# set protocols mvrp xe-0/1/0.0
   ```

NOTE: As we recommend as a best practice, default MVRP timers are used in this example. 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.

Results Check the results of the configuration for Switch B:

```
[edit]
user@Access-Switch-B# show
interfaces {
    ge-0/0/0 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members finance;
                }
            }
        }
    }
    ge-0/0/1 {
        unit 0 {
            family ethernet-switching {
                vlan {
                    members lab;
                }
            }
        }
    }
    xe-0/1/0 {
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
            }
        }
    }
}

protocols {
    mvrp {
        interface xe-0/1/0.0;
    }
}
vlans {
    finance {
        vlan-id 100;
    }
    lab {
        vlan-id 200;
    }
    sales {
```

```
    vlan-id 300;
    }
  }
```

Configuring VLANS and MVRP on Distribution Switch C

CLI Quick Configuration

To quickly configure distribution Switch C for MVRP, copy the following commands and paste them into the switch terminal window of distribution Switch C:

```
[edit]
set interfaces xe-0/1/1 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/1/0 unit 0 family ethernet-switching port-mode trunk
set protocols mvrp interface xe-0/1/1.0
set protocols mvrp interface xe-0/1/0.0
```

Step-by-Step Procedure

To configure distribution Switch C for MVRP:

1. Configure the trunk interface to access Switch A:

   ```
   [edit]
   user@Distribution-Switch-C# set interfaces xe-0/1/1 unit 0 family ethernet-switching
   port-mode trunk
   ```

2. Configure the trunk interface to access Switch B:

   ```
   [edit]
   user@Distribution-Switch-C# set interfaces xe-0/1/0 unit 0 family ethernet-switching
   port-mode trunk
   ```

3. Enable MVRP on the trunk interface for **xe-0/1/1** :

   ```
   [edit]
   user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/1.0
   ```

4. Enable MVRP on the trunk interface for **xe-0/1/0** :

   ```
   [edit]
   user@Distribution-Switch-C# set protocols mvrp interface xe-0/1/0.0
   ```

Results

Check the results of the configuration for Switch C:

```
[edit]
user@Distribution Switch-C# show
interfaces {
  xe-0/1/0 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
      }
    }
  }
  xe-0/1/1 {
    unit 0 {
```

```
        family ethernet-switching {
          port-mode trunk;
        }
      }
    }
  }
  protocols {
    mvrp {
      interface xe-0/1/0.0;
      interface xe-0/1/1.0;
    }
```

Verification

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

- Verifying That MVRP Is Enabled on Access Switch A on page 545
- Verifying That MVRP Is Updating VLAN Membership on Access Switch A on page 546
- Verifying That MVRP Is Enabled on Access Switch B on page 546
- Verifying That MVRP Is Updating VLAN Membership on Access Switch B on page 546
- Verifying That MVRP Is Enabled on Distribution Switch C on page 547
- Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C on page 547

Verifying That MVRP Is Enabled on Access Switch A

Purpose Verify that MVRP is enabled on the switch.

Action Show the MVRP configuration:

```
user@Access-Switch-A> show mvrp
MVRP configuration
MVRP status                 : Enabled
MVRP dynamic VLAN creation  : Enabled

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

Interface       Status        Registration Mode
--------------  --------      -----------------
all             Disabled      Normal
xe-0/1/1.0      Enabled       Normal
```

Meaning The results show that MVRP is enabled on the trunk interface of Switch A and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch A

Purpose Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch A.

Action List Ethernet switching interfaces on the switch:

```
user@Access-Switch-A> show ethernet-switching interfaces
Interface   State    VLAN members   Tag  Tagging       Blocking
ge-0/0/1.0  up       finance        100  untagged      unblocked
ge-0/0/2.0  up       lab            200  untagged      unblocked
ge-0/0/3.0  up       sales          300  untagged      unblocked
xe-0/1/1.0  up       finance        100  untagged      unblocked
                     lab            200  untagged      unblocked
```

Meaning MVRP has automatically added **finance** and **lab** as VLAN members on the trunk interface because they are being advertised by access Switch B.

Verifying That MVRP Is Enabled on Access Switch B

Purpose Verify that MVRP is enabled on the switch.

Action Show the MVRP configuration:

```
user@Access-Switch-B> show mvrp

MVRP configuration
MVRP status                 : Enabled
MVRP dynamic VLAN creation  : Enabled

MVRP timers (ms):
Interface          Join   Leave       LeaveAll
--------------     -----  --------     -----------
all                 200   1000        10000
xe-0/1/0.0          200   1000        10000

Interface       Status        Registration Mode
--------------  --------      -----------------
all             Disabled      Normal
xe-0/1/0.0      Enabled       Normal
```

Meaning The results show that MVRP is enabled on the trunk interface of Switch B and that the default timers are used.

Verifying That MVRP Is Updating VLAN Membership on Access Switch B

Purpose Verify that MVRP is updating VLAN membership by displaying the Ethernet switching interfaces and associated VLANs that are active on Switch B.

Action List Ethernet switching interfaces on the switch:

```
user@Access-Switch-B> show ethernet-switching interfaces
Interface   State    VLAN members   Tag  Tagging       Blocking
ge-0/0/0.0  up       finance        100  untagged      unblocked
ge-0/0/1.0  up       lab            200  untagged      unblocked
xe-0/1/1.0  up       finance        100  untagged      unblocked
                     lab            200  untagged      unblocked
                     sales          300  untagged      unblocked
```

Meaning MVRP has automatically added **finance**, **lab**, and **sales** as VLAN members on the trunk interface because they are being advertised by access Switch A.

Verifying That MVRP Is Enabled on Distribution Switch C

Purpose Verify that MVRP is enabled on the switch.

Action Show the MVRP configuration:

```
user@Distribution-Switch-C> show mvrp

MVRP configuration
MVRP status                 : Enabled
MVRP dynamic VLAN creation  : Enabled

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

Interface       Status        Registration Mode
--------------  --------      -----------------
all             Disabled      Normal
xe-0/0/1.0      Enabled       Normal
xe-0/1/1.0      Enabled       Normal
```

Verifying That MVRP Is Updating VLAN Membership on Distribution Switch C

Purpose Verify that MVRP is updating VLAN membership on distribution Switch C by displaying the Ethernet switching interfaces and associated VLANs on distribution Switch C.

Action List the Ethernet switching interfaces on the switch:

```
user@Distribution-Switch-C> show ethernet-switching interfaces
Interface   State    VLAN members   Tag  Tagging       Blocking
xe-0/1/1.0  up       __mvrp_100__                    unblocked
                     __mvrp_200__                    unblocked
                     __mvrp_300__                    unblocked
xe-0/1/0.0  up       __mvrp_100__                    unblocked
                     __mvrp_200__                    unblocked
```

List the VLANs that were created dynamically using MVRP on the switch:

```
user@Distribution-Switch-C> show mvrp dynamic-vlan-memberships

MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN ID          Interfaces
100              xe-0/1/1.0
                 xe-0/1/0.0
200              xe-0/1/1.0
                 xe-0/1/0.0
300              xe-0/1/1.0
```

Note that this scenario does not have any fixed registration, which is typical when MVRP is enabled.

Meaning Distribution Switch C has two trunk interfaces. Interface **xe-0/1/1.0** connects distribution Switch C to Access Switch A and is therefore updated to show that it is a member of all the VLANs that are active on Switch A. Any traffic for those VLANs will be passed on from distribution Switch C to Switch A, through interface **xe-0/1/1.0**. Interface **xe-0/1/0.0** connects distribution Switch C to Switch B and is updated to show that it is a member of the two VLANs that are active on Switch B. Thus, distribution Switch C sends traffic for **finance** and **lab** to both Switch A and Switch B. But distribution Switch C sends traffic for **sales** only to Switch A.

Distribution Switch C also has three dynamic VLANs created using MVRP: **mvrp_100**, **mvrp_200**, and **mvrp_300**. The dynamically created VLANs **mvrp_100** and **mvrp_200** are active on interfaces **xe-0/1/1.0** and **xe-0/1/1.0**, and dynamically created VLAN **mvrp_300** is active on interface **xe-0/1/1.0**.

Related Documentation

- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496

Verifying That MVRP Is Working Correctly on Switches

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) on Switches" on page 504.

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535

- Example: Configuring Automatic VLAN Administration on QFX Switches Using MVRP on page 516
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504

Verifying That MVRP Is Working Correctly on EX Series Switches with ELS Support

Purpose

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Verifying That MVRP Is Working Correctly on Switches" on page 548. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

After configuring your EX Series 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
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  xe-0/1/1         200    1000     10000
```

2. Confirm that MVRP messages are being sent and received on your switch.

```
user@switch> show mvrp statistics
MVRP statistics for routing instance 'default-switch'

Interface name                  : xe-0/1/1
VLAN IDs registered             : 117
Sent MVRP PDUs                  : 118824
Received MVRP PDUs without error: 118848
Received MVRP PDUs with error   : 0
Transmitted Join Empty          : 5229
Transmitted Leave All           : 2
Recieved Join In                : 11884924
Transmitted Join In             : 1835
Transmitted Empty               : 93606408
Transmitted Leave               : 888
Transmitted In                  : 13780024
Transmitted New                 : 2692
Received Leave All              : 118761
Received Leave                  : 97
Received In                     : 3869
Received Empty                  : 828
Received Join Empty             : 2020152
Received New                    : 224
...
```

Meaning The output of **show mvrp** shows that interface xe-0/1/1 is enabled for MVRP participation.

The output for **show mvrp statistics** confirms that MVRP messages are being transmitted and received on interface xe-0/1/1.

NOTE: You can identify an MVRP compatibility issue by observing the output from this command. If **Received Join Empty** and **Received Join In** incorrectly display zero, even though the value for **Received MVRP PDUs without error** has been increased, you are probably running different versions of Junos OS 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) on Switches" on page 504.

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support on page 521
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496

Verifying That MVRP Is Working Correctly

Purpose After configuring your MX Series router or EX Series switch to participate in Multiple VLAN Registration Protocol (MVRP), verify that the configuration is properly set and that MVRP messages are being sent and received on your switch.

Action

1. Confirm that the router is declaring VLANs.

   Show that MVRP is enabled:

```
user@host> show mvrp
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  ge-11/3/0        200     800     10000
```

   Show the MVRP applicant state:

```
user@host> show mvrp applicant-state
MVRP applicant state for routing instance 'default-switch'
 (VO) Very anxious observer, (VP) Very anxious passive, (VA) Very anxious new,

 (AN) Anxious new, (AA) Anxious active, (QA) Quiet active, (LA) Leaving active,

 (AO) Anxious observer, (QO) Quiet observer, (LO) Leaving observer,
 (AP) Anxious passive, (QP) Quiet passive
```

```
VLAN Id     Interface          State
    100     ge-11/3/0          Declaring (QA)
    200     ge-11/3/0          Declaring (QA)
    300     ge-11/3/0          Declaring (QA)
```

2. Confirm that VLANs are registered on interfaces.

   List VLANs in the registered state:

```
user@host> show mvrp registration-state
MVRP registration state for routing instance 'default-switch'

VLAN Id   Interface   Registrar   Forced      Managed    STP
                      State       State       State      State
    100   ge-11/3/0   Registered  Registered  Normal     Forwarding
    200   ge-11/3/0   Registered  Registered  Normal     Forwarding
    300   ge-11/3/0   Empty       Empty       Normal     Forwarding
```

3. Display a list of VLANs created dynamically.

   List dynamic VLAN membership:

```
user@host> show mvrp dynamic-vlan-memberships
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN Id            Interfaces
    100            ge-3/3/0
                   ge-3/0/5
    200            ge-3/3/0
                   ge-3/0/5
```

Meaning

The output of **show mvrp applicant-state** shows that trunk interface **ge-11/3/0** is declaring (sending out) interest in the VLAN IDs **100**, **200**, and **300**, and MVRP is operating properly.

The output of **show mvrp registrant-state** shows the registrar state for VLANs **100** and **200** as **Registered**, indicating that these VLANs are receiving traffic from a customer site. VLAN **300** is in an **Empty** state and is not receiving traffic from a customer site.

The output of the **show mvrp dynamic-vlan-membership** shows that VLANs **100** and **200** are created dynamically (here, on an MX Series router operating as an aggregation switch between MX Series routers operating as edge switches). VLANs created statically are marked with an **(s)** (which is not indicated in this output).

Related Documentation

- *Controlling the Management State of a VLAN in MVRP Configurations*

CHAPTER 21

Configuring Q-in-Q Tunneling and VLAN Translation

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.

Using Q-in-Q tunneling, providers can segregate or bundle customer traffic into fewer VLANs or different VLANs by adding another layer of 802.1Q tags. Q-in-Q tunneling is useful when customers have overlapping VLAN IDs, because the customer's 802.1Q (dot1Q) VLAN tags are prepended by the service VLAN (S-VLAN) tag. The Juniper Networks Junos operating system (Junos OS) implementation of Q-in-Q tunneling supports the IEEE 802.1ad standard.

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 VLAN, a customer-specific 802.1Q tag is added to the packet. This additional tag is used to segregate traffic into service-provider-defined service VLANs (S-VLANs). The original customer 802.1Q tag of the packet remains and is transmitted transparently, passing through the service provider's network. As the packet leaves the S-VLAN in the downstream direction, the extra 802.1Q tag is removed.

NOTE: **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.**

When Q-in-Q tunneling is enabled on Juniper Networks EX Series Ethernet Switches, trunk interfaces are assumed to be part of the service provider network and access interfaces are assumed to be customer facing. An access interface can receive both tagged and untagged frames in this case.

NOTE: Starting with Junos OS 14.1X53-D30, you can configure the same interface to be an S-VLAN/NNI interface and a C-VLAN/UNI interface. This means that the same physical interface can transmit single-tagged and double-tagged frames simultaneously. This allows you maximum flexibility in your network topology and lets you maximize the use of your interfaces.

An interface can be a member of multiple S-VLANs. You can map one C-VLAN to one S-VLAN (1:1) or multiple C-VLANs to one S-VLAN (N:1). Packets are double-tagged for an additional layer of segregating or bundling of C-VLANs. C-VLAN and S-VLAN tags are unique; 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. Class-of-service (CoS) values of C-VLANs are unchanged in the downstream direction. You may, optionally, copy ingress priority and CoS settings to the S-VLAN. On non-ELS Switches, you can use private VLANs to isolate users to prevent the forwarding of traffic between user interfaces even if the interfaces are on the same VLAN.

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.)

NOTE: Priority tagged packets are not supported with Q-in-Q tunneling on QFX5100 and EX4600 switches.

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.

You can use the **native** option to specify an S-VLAN for untagged and priority tagged packets when using many-to-one bundling and mapping a specific interface approaches to map C-VLANs to S-VLANs. (This does not apply to switches supporting ELS.) Otherwise the 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 the S-VLAN. See the Mapping C-VLANs to S-VLANs section of this document for information on the methods of mapping C-VLANs to S-VLANs.

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.

On non-ELS switches, firewall filters allow you to map an interface to a VLAN based on a policy. Using firewall filters to map an interface to a VLAN is useful when you want a subset of traffic from a port to be mapped to a selected VLAN instead of the designated VLAN. To configure a firewall filter to map an interface to a VLAN, the **vlan** option has to be configured as part of the firewall filter and the **mapping policy** option must be specified in the interface configuration for each logical interface using the filter.

NOTE: On an EX4300 switch, you can configure multiple logical interfaces on the same Ethernet port, but each logical interface supports only single-tagged packets and that tag must include a different VLAN ID than those supported by the other logical interfaces. Given this situation, you cannot enable Q-in-Q tunneling on Ethernet ports with multiple logical subinterfaces.

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. Incoming packets whose tags do not match the C-VLAN tag are dropped, unless additional VLAN translation configuration for those tags exist.

To configure VLAN translation, use the mapping **swap** statement at the **[edit vlans interface]** hierarchy level. As long as the C-VLAN and S-VLAN tags are unique, you can configure more than one C-VLAN-to-S-VLAN translation on an access port. If you are translating only one VLAN on an interface, you do not need to include the **dot1q-tunneling**

statement in the S-VLAN configuration. If you are translating more than one VLAN, you must use the **dot1q-tunneling** statement.

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. You can configure only 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.

NOTE: VLAN translation is not supported on QFabric systems.

Using Dual VLAN Tag Translation

Starting with Junos OS Release 14.1X53-D40, you can use the dual VLAN tag translation (also known as dual VLAN tag rewrite) feature to deploy switches in service-provider domains, allowing dual-tagged, single-tagged, and untagged VLAN packets to come into or exit from the switch. Table 87 on page 557 shows the operations that are added for dual VLAN tag translation.

Table 87: Operations Added with Dual VLAN Tag Rewrite

Operation	Function
swap-push	Swap a VLAN tag and push a new VLAN tag
pop-swap	Pop an outer VLAN tag and swap an inner VLAN tag
swap-swap	Swap both outer and inner VLAN tags

Dual VLAN tag translation supports:

- Configuration of S-VLANs (NNI) and C-VLANs (UNI) on the same physical interface
- Control protocols such as VSTP, OSPF, and LACP
- IGMP snooping
- Configuration of a private VLAN (PVLAN) and VLAN on a single-tagged interface
- Use of TPID 0x8100 on both inner and outer VLAN tags

See "Setting Up a Dual VLAN Tag Translation Configuration on QFX Switches" on page 605.

Sending and Receiving Untagged Packets

To enable an interface to send and receive untagged packets, you must specify a native VLAN for a physical interface. When the interface receives an untagged packet, it adds the VLAN ID of the native VLAN to the packet and sends the newly tagged packet to the mapped interface.

To specify a native VLAN, use the **native-vlan-id** statement at the **[edit interfaces *interface-name*]** hierarchy level. The native VLAN ID must match the C-VLAN or S-VLAN ID or be included in the VLAN ID list specified on the logical interface.

For example, on a logical interface for a C-VLAN interface, you might specify a C-VLAN ID list of 100-200. Then, on the C-VLAN physical interface, you could specify a native VLAN ID of 150. This configuration would work because the native VLAN of 150 is included in the C-VLAN ID list of 100-200.

We recommend configuring a native VLAN when using any of the approaches to map C-VLANs to S-VLANs. If you do not configure a native VLAN on an interface, untagged packets received by the interface are discarded. See the Mapping C-VLANs to S-VLANs section in this topic for information about the methods of mapping C-VLANs to S-VLANs.

Disabling MAC Address Learning

In a Q-in-Q deployment, customer packets from downstream interfaces are transported without any changes to source and destination MAC addresses. You can disable MAC address learning at global, interface, and VLAN levels:

- To disable learning globally, disable MAC address learning for the switch.
- To disable learning for an interface, disable MAC address learning for all VLANs of which the specified interface is a member.
- To disable learning for a VLAN, disable MAC address learning for a specified VLAN.

Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member. When you disable MAC address learning on a VLAN, MAC addresses that have already been learned are flushed.

If you disable MAC address learning on an interface or a VLAN, you cannot include 802.1X authentication in that same VLAN configuration.

When a routed VLAN interface (RVI) is associated with either an interface or a VLAN on which MAC address learning is disabled, the Layer 3 routes resolved on that VLAN or that interface are not resolved with the Layer 2 component. This results in routed packets flooding all the interfaces associated with the VLAN.

Mapping C-VLANs to S-VLANs

There are multiple ways to map C-VLANs to an S-VLAN:

NOTE: If you configure multiple mapping methods, the switch gives priority to mapping a specific interface, then to many-to-many bundling, and last to all-in-one bundling. However, for a particular mapping method, setting up overlapping rules for the same C-VLAN is not supported.

- 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.)
- Many-to-many bundling—Use many-to-many bundling when you want a subset of the C-VLANs on the access switch to be part of multiple S-VLANs.
- 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 gives priority to mapping a specific interface, then to many-to-one bundling, and last to all-in-one bundling. However, you cannot have overlapping rules for the same C-VLAN under a given approach. For example, you cannot use many-to one bundling to map C-VLAN 100 to two different S-VLANs.

- All-in-One Bundling on page 559
- Many-to-One Bundling on page 560
- Many-to-Many Bundling on page 560
- Mapping a Specific Interface on page 560
- Combining Methods and Configuration Restrictions on page 561

All-in-One Bundling

All-in-one bundling maps all packets from all C-VLAN interfaces to an S-VLAN.

The C-VLAN interface accepts untagged and single-tagged packets. An S-VLAN 802.1Q tag is then added to these packets, and the packets are sent to the S-VLAN interface, which accepts untagged, single-tagged, and double-tagged packets.

NOTE: The C-VLAN and S-VLAN interfaces accept untagged packets provided that the **native-vlan-id** statement is configured on these interfaces.

Many-to-One Bundling

Many-to-one bundling is used to specify which C-VLANs are mapped to an S-VLAN. Many-to-one bundling is configured using the **customer-vlans** option.

Many-to-one bundling is used when you want a subset of the C-VLANs on the access switch to be part of the S-VLAN. When using many-to-one bundling, untagged and priority tagged packets can be mapped to the S-VLAN when the **native** option is specified along with the **customer-vlans** option.

Many-to-Many Bundling

Many-to-many bundling is used to specify which C-VLANs are mapped to which S-VLANs.

Use many-to-many bundling when you want a subset of the C-VLANs on the access switch to be part of multiple S-VLANs. With many-to-many bundling, the C-VLAN interfaces accept untagged and single-tagged packets. An S-VLAN 802.1Q tag is then added to these packets, and the packets are sent to the S-VLAN interfaces, which accept untagged, single-tagged, and double-tagged packets.

NOTE: The C-VLAN and S-VLAN interfaces accept untagged packets provided that the **native-vlan-id** statement is configured on these interfaces.

Mapping a Specific Interface

Use specific interface mapping when you want to assign an S-VLAN to a specific C-VLAN on an interface. The configuration applies only to the specific interface, not to all access interfaces.

Specific interface mapping has two suboptions: **push** and **swap**. When traffic that is mapped to a specific interface is pushed, the packet retains its original tag as it moves from the C-VLAN to the S-VLAN and an additional S-VLAN tag is added to the packet. When traffic that is mapped to a specific interface is swapped, the incoming tag is replaced with a new VLAN tag. This is sometimes known VLAN rewriting or VLAN translation.

Typically, this method is used to keep data from different customers separate or to provide individualized treatment of the packets on a certain interface. You might also use this method map VLAN traffic from different customers to a single S-VLAN.

When using specific interface mapping, the C-VLAN interfaces accept untagged and single-tagged packets, while the S-VLAN interfaces accept untagged, single-tagged, and double-tagged packets.

NOTE: The C-VLAN and S-VLAN interfaces accept untagged packets provided that the **native-vlan-id** statement is configured on these interfaces.

Combining Methods and Configuration Restrictions

If you configure multiple methods, the switch gives priority to mapping a specific interface, then to many-to-one bundling, and last to all-in-one bundling. An access interface configured under all-in-one bundle cannot be part of a many-to-one bundle. It can have additional mappings defined, however.

To ensure deterministic results, the following configuration restrictions apply:

- Mapping cannot be defined for untagged vlans.
- An access interface can have multiple customer VLAN ranges, but an interface cannot have overlapping tags across the VLANs.

  For example, the following configuration is not allowed:

```
vlans {
    customer-1 {
        vlan-id 100;                    /* S-VLAN */
        interfaces ge-0/0/0.0;          /* Downstream */
        interfaces ge-0/0/1.0;          /* Downstream */
        interfaces xe-0/1/0.0;          /* trunk */
        dot1q-tunnelling customer-vlans 100-200 300-400
        }
    }
    customer-2 {
        vlan-id 200;
        interfaces ge-0/0/0.0;          /* Downstream */
        interfaces xe-0/1/0.0;          /* trunk */
        dot1q-tunnelling customer-vlans 250-350
        }
    }
    customer-3 {
        vlan-id 300;
        interfaces ge-0/0/1.0;          /* Downstream */
        interfaces xe-0/1/0.0;          /* trunk */
        dot1q-tunnelling customer-vlans 500-600
    }
}
```

  Because the **customer-2** configuration creates overlapping **customer-vlan** ranges for ge-0/0/0, it is invalid.

- An access interface can have a single rule that maps an untagged packet to a VLAN.
- Each interface can have at most one mapping swap rule per VLAN.
- You can push a VLAN tag only on the access ports of a Q-in-Q VLAN. This restriction applies to all three methods of pushing a VLAN tag: that is, all-in-one bundling, many-to-one-bunding, and mapping a specific interface using push.
- You can push different C-VLAN tags for a given S-VLAN on different interfaces. This could potentially result in traffic leaking across VLANs, depending on your configuration.

Routed VLAN Interfaces on Q-in-Q VLANs

Routed VLAN interfaces (RVIs) are supported on Q-in-Q VLANs.

Packets arriving on an RVI that is using Q-in-Q VLANs will get routed regardless of whether the packet is single or double tagged. The outgoing routed packets contain an S-VLAN tag only when exiting a trunk interface; the packets exit the interface untagged when exiting 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:

- Q-in-Q tunneling supports only two VLAN tags.
- Q-in-Q tunneling does not support most access port security features. There is no per-VLAN (customer) policing or per-VLAN (outgoing) shaping and limiting with Q-in-Q tunneling unless you configure these security features by using firewall filters.
- With releases of Junos OS Release 13.2X51 previous to Release 13.2X51-D20, you cannot create a regular VLAN on an interface if you have created an S-VLAN or C-VLAN on that interface for Q-in-Q tunneling. This means that you cannot create an integrated routing and bridging (IRB) interface on that interface because regular VLANs are a required part of IRB configuration. With Junos OS Release 13.2X51-D25, you can create a regular VLAN on a trunk interface that has an S-VLAN, which means that you can also create an IRB interface on the trunk. In this case, the regular VLAN and S-VLAN on the same trunk interface cannot share the same VLAN ID. Junos OS Release 13.2X51-D25 does not allow you to create a regular VLAN on an access interface that has a C-VLAN.
- Starting with Junos OS Release 14.1X53-D40, integrated routing and bridging(IRB) interfaces are supported on Q-in-Q VLANs—you can configurethe IRB interface on the same interface as one used by an S-VLAN,and you can use the same VLAN ID for both the VLAN used by the IRBinterface and for the VLAN used as an S-VLAN.

  Packets arriving on an IRB interface that is using Q-in-Q VLANswill get routed regardless of whether the packet is single taggedor double tagged. The outgoing routed packets contain an S-VLAN tagonly when exiting a trunk interface; the packets exit the interface untagged when exiting an access interface.

  NOTE: **You can configure the IRB interface only on S-VLAN (NNI) interfaces, not on C-VLAN (UNI) interfaces.**
- Most access port security features are not supported with Q-in-Q tunneling and VLAN translation.
- Configuring Q-in-Q tunneling and VLAN rewriting/VLAN translation on the same port is not supported.

- You can configure at most one VLAN rewrite/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.
- You cannot use the native VLAN ID
- 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 rewriting/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

Release History Table

Release	Description
14.1X53-D30	Starting with Junos OS 14.1X53-D30, you can configure the same interface to be an S-VLAN/NNI interface and a C-VLAN/UNI interface.

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- Configuring Q-in-Q Tunneling on page 564
- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582
- Configuring Q-in-Q Tunneling on QFX Series Switches on page 581
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598

- *Troubleshooting Q-in-Q and VLAN Translation Configuration*
- *mtu*

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 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.

NOTE: **This task uses a Junos OS release that supports the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see** "Configuring Q-in-Q Tunneling on QFX Series Switches" on page 581.

With releases of Junos OS 13.2X51 previous to 13.2X51-D20, you cannot create a regular VLAN on an interface if you have created an S-VLAN or C-VLAN on that interface for Q-in-Q tunneling. This means that you cannot create an integrated routing and bridging (IRB) interface on that interface because regular VLANs are a required part of IRB configuration. With Junos OS 13.2X51-D25, you can create a regular VLAN on a trunk interface that has an S-VLAN, which means that you can also create an IRB interface on the trunk. In this case, the regular VLAN and S-VLAN on the same trunk interface cannot share the same VLAN ID. Junos OS 13.2X51-D25 does not allow you to create a regular VLAN on an access interface that has a C-VLAN.

Before 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 Switches" on page 93.

Using the Different Mapping Methods

Once you have created the required VLANs on the neighboring switches, configure Q-in-Q tunneling using one of the three methods to map customer VLANs (C-VLANs) to service-provider-defined service VLANs (S-VLANs):.

- All-in-one bundling maps all packets from all C-VLAN interfaces to an S-VLAN. For information about how to use this method, see "Configuring Q-in-Q Tunneling Using All-in-One Bundling" on page 565.
- Use many-to-many bundling when you want a subset of the C-VLANs on the access switch to be part of multiple S-VLANs. For information about how to use this method, see "Configuring Q-in-Q Tunneling Using Many-to-Many Bundling" on page 567.
- Use specific interface mapping when you want to assign an S-VLAN to a specific C-VLAN on an interface. For information about how to use this method, see "Configuring a Specific Interface Mapping with VLAN ID Translation Option" on page 570.

Configuring Q-in-Q Tunneling Using All-in-One Bundling

You can configure Q-in-Q tunneling using the all-in-one bundling method, which forwards all packets that ingress on a C-VLAN interface to an S-VLAN. (Packets are forwarded to the S-VLAN regardless of whether they are tagged or untagged prior to ingress.) Using this approach saves you the effort of specifying a specific mapping for each C-VLAN.

First configure the S-VLAN and its interface:

1. Assign a logical interface (unit) to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **interface *interface-name.unit-number***

   NOTE: Do not use logical interface unit 0. You must later bind a VLAN tag ID to the unit you specify in this step, and you cannot bind a VLAN tag ID to unit 0. Also note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Enable the interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

   NOTE: If you configure an enterprise-style configuration such as PVLAN on the same physical interface on which you are configuring Q-in-Q tunneling, use **set encapsulation flexible-ethernet-services** in step 3. See *Understanding Flexible Ethernet Services Encapsulation on Switches*.

4. Enable the S-VLAN interface to send and receive untagged packets:

```
[edit interfaces interface-name]
user@switch# set native-vlan-id vlan-id
```

5. Bind the logical interface (unit) of the interface that you specified in step 1 to the automatically created VLAN ID for the S-VLAN:

```
[edit interfaces interface-name unit logical-unit-number]
user@switch# set vlan-id number
```

NOTE: If you configured flexible-ethernet-services in step 3, configure vlan-bridge encapsulation on the logical interface:

```
[edit interfaces interface-name unit logical-unit-number]
user@switch# set encapsulation vlan-bridge
```

For example, the following configuration makes xe-0/0/0.10 a member of VLAN 10, enables Q-in-Q tunneling on interface xe-0/0/0, enables xe-0/0/0 to accept untagged packets, and binds the VLAN ID of S-VLAN v10 to a logical interface of xe-0/0/0.

```
set vlans v10 interface xe-0/0/0.10
set interfaces xe-0/0/0 flexible-vlan-tagging
set interfaces xe-0/0/0 native-vlan-id 10
set interfaces xe-0/0/0 encapsulation extended-vlan-bridge
set interfaces xe-0/0/0 unit 10 vlan-id 10
```

Now configure all-in-one bundling on a C-VLAN interface:

1. Assign a logical interface (unit) of the C-VLAN interface to be a member of the S-VLAN.

```
[edit vlans vlan-name]
user@switch# set interface interface-name.unit-number
```

2. Enable the interface to transmit packets with 802.1Q VLAN tags :

```
[edit interfaces interface-name]
user@switch# set flexible-vlan-tagging
```

3. Enable extended VLAN bridge encapsulation on the interface:

```
[edit interfaces interface-name]
user@switch# set encapsulation extended-vlan-bridge
```

4. Enable the C-VLAN interface to send and receive untagged packets:

```
[edit interfaces interface-name]
user@switch# set native-vlan-id vlan-id
```

5. Configure a logical interface to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

```
[edit interfaces interface-name unit logical-unit-number]
user@switch# set vlan-id-list vlan-id-numbers
```

CAUTION: You can apply no more than eight VLAN identifier lists to a physical interface. This limitation does not apply to QFX10000 switches.

6. Configure the system to add an S-VLAN tag (outer tag) as packets travel from a C-VLAN interface to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** input-vlan-map push

NOTE: You can configure vlan-id on input-vlan-map, but doing so is optional.

7. Configure the system to remove the S-VLAN tag when packets are forwarded (internally) from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** output-vlan-map pop

For example, the following configuration makes xe-0/0/1.10 a member of S-VLAN v10, enables Q-in-Q tunneling, maps packets from C-VLANs 100 through 200 to S-VLAN 10, and enables xe-0/0/1 to accept untagged packets. If a packet originates in C-VLAN 100 and needs to be sent across the S-VLAN, a tag with VLAN ID 10 is added to the packet. When a packet is forwarded (internally) from the S-VLAN interface to interface xe-0/0/1, the tag with VLAN ID 10 is removed.

```
set vlans v10 interface xe-0/0/1.10
set interfaces xe-0/0/1 flexible-vlan-tagging
set interfaces xe-0/0/1 encapsulation extended-vlan-bridge
set interfaces xe-0/0/1 unit 10 vlan-id-list 100-200
set interfaces xe-0/0/1 native-vlan-id 150
set interfaces xe-0/0/1 unit 10 input-vlan-map push
set interfaces xe-0/0/1 unit 10 output-vlan-map pop
```

See Also

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using Many-to-Many Bundling on page 567
- Configuring a Specific Interface Mapping with VLAN ID Translation Option on page 570

Configuring Q-in-Q Tunneling Using Many-to-Many Bundling

You can configure Q-in-Q tunneling using the many-to-many bundling method, which maps packets from multiple C-VLANs to multiple S-VLANs. This method is convenient for mapping a range of C-VLANs without having to specify each one individually. (You can also use this method to configure only one C-VLAN to be mapped to an S-VLAN.)

First configure the S-VLANs and assign them to an interface:

1. Assign a logical interface (unit) to be a member of one of the S-VLANs. Do not use logical interface unit 0.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

   NOTE: Note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Repeat step 1 for the other S-VLANs.

3. Enable the physical interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

5. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

6. Bind one of the logical units of the interface to the VLAN ID for one of the S-VLANs.

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

7. Repeat step 6 to bind the automatically-created VLAN IDs for the other S-VLANs to the other logical units of the interface:

For example, the following configuration creates S-VLANs v10 and v30 and associates them with interface xe-0/0/0.10, enables Q-in-Q tunneling, enables xe-0/0/0 to accept untagged packets, and maps incoming C-VLAN packets to S-VLANs v10 and v30.

```
set vlans v10 interface xe-0/0/0.10
set vlans v30 interface xe-0/0/0.10
set interfaces xe-0/0/0 flexible-vlan-tagging
set interfaces xe-0/0/0 native-vlan-id 10
set interfaces xe-0/0/0 encapsulation extended-vlan-bridge
set interfaces xe-0/0/0 unit 10 vlan-id 10
set interfaces xe-0/0/0 unit 30 vlan-id 30
```

To configure the many-to-many bundling method on a C-VLAN interface, perform the following steps for each customer:

1. Assign a logical interface (unit) of one C-VLAN interface to be a member of one S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

2. Repeat step 1 to assign another C-VLAN interface (physical interface) to be a member of another S-VLAN.

3. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set** flexible-vlan-tagging

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` encapsulation **extended-vlan-bridge**

5. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

6. For each physical interface, configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id-list *vlan-id-numbers***

   To configure only one C-VLAN to be mapped to an S-VLAN, specify only one VLAN ID after *vlan-id-list*.

   CAUTION: You can apply no more than eight VLAN identifier lists to a physical interface. This limitation does not apply to QFX10000 switches.

7. For each physical interface, configure the system to add an S-VLAN tag (outer tag) as packets travel from the C-VLAN interface to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** input-vlan-map push

8. For each physical interface, configure the system to remove the S-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** output-vlan-map pop

For example, the following configuration makes xe-0/0/1.10 a member of S-VLAN v10, enables Q-in-Q tunneling, and maps packets from C-VLANs 10 through 20 to S-VLAN 10. The configuration for customer 2 makes xe-0/0/2.30 a member of S-VLAN v30, enables Q-in-Q tunneling, and maps packets from C-VLANs 30 through 40, 50 through 60, and 70 through 80 to S-VLAN 30. Both interfaces are configured to accept untagged packets.

If a packet originates in C-VLAN 10 and needs to be sent over the S-VLAN, a tag with a VLAN ID 10 is added to the packet. If a packet is forwarded internally from the S-VLAN interface to xe-0/0/1.10, the tag with VLAN ID 10 is removed. The same principles apply to the C-VLANs configured on interface xe-0/0/2.

NOTE: Notice that you can use the same tag value for an S-VLAN and C-VLAN. For example, the configuration for customer 1 maps C-VLAN ID 10 to S-VLAN ID 10. C-VLAN and S-VLAN tags use separate name spaces, so this configuration is allowed.

Configuration for customer 1:

```
set vlans v10 interface xe-0/0/1.10
set interfaces xe-0/0/1 flexible-vlan-tagging
set interfaces xe-0/0/1 encapsulation extended-vlan-bridge
set interfaces xe-0/0/1 unit 10 vlan-id-list 10-20
set interfaces xe-0/0/1 native-vlan-id 15
set interfaces xe-0/0/1 unit 10 input-vlan-map push
set interfaces xe-0/0/1 unit 10 output-vlan-map pop
```

Configuration for customer 2:

```
set vlans v30 interface xe-0/0/2.30
set interfaces xe-0/0/2 flexible-vlan-tagging
set interfaces xe-0/0/2 encapsulation extended-vlan-bridge
set interfaces xe-0/0/2 unit 30 vlan-id-list 30-40
set interfaces xe-0/0/2 unit 30 vlan-id-list 50-60
set interfaces xe-0/0/2 unit 30 vlan-id-list 70-80
set interfaces xe-0/0/2 native-vlan-id 75
set interfaces xe-0/0/2 unit 30 input-vlan-map push
set interfaces xe-0/0/2 unit 30 output-vlan-map pop
```

See Also

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using All-in-One Bundling on page 565
- Configuring a Specific Interface Mapping with VLAN ID Translation Option on page 570

Configuring a Specific Interface Mapping with VLAN ID Translation Option

You can configure Q-in-Q tunneling by mapping packets from a specified C-VLAN to a specified S-VLAN. In addition, you can configure the system to replace a C-VLAN tag with an S-VLAN tag or replace an S-VLAN tag with a C-VLAN tag (instead of double tagging). This is call VLAN translation or VLAN rewriting. VLAN translation is particularly

useful if a service provider's Layer 2 network that connects a customer's sites does not support double tagged packets.

When you use VLAN translation, both ends of the link normally must be able to swap the tags appropriately. That is, both ends of the link must be configured to swap the C-VLAN tag for the S-VLAN tag and swap the S-VLAN tag for the C-VLAN tag so that traffic in both directions is tagged appropriately while in transit and after arrival.

First configure the S-VLAN and its interface:

1. Assign a logical interface to be a member of the S-VLAN. Do not use unit 0.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

   NOTE: Note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

5. Bind the logical interface (unit) of the interface that you specified earlier to the VLAN ID for the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

For example, the following configuration creates S-VLAN v200, makes xe-0/0/0.200 a member of that VLAN, enables Q-in-Q tunneling on interface xe-0/0/0, enables xe-0/0/0 to accept untagged packets, and binds a logical interface of xe-0/0/0 to the VLAN ID of VLAN v200.

```
set vlans v200 interface xe-0/0/0.200
set interfaces xe-0/0/0 flexible-vlan-tagging
set interfaces xe-0/0/0 native-vlan-id 150
set interfaces xe-0/0/0 encapsulation extended-vlan-bridge
set interfaces xe-0/0/0 unit 200 vlan-id 200
```

Now configure a specific interface mapping with optional VLAN ID translation on the C-VLAN interface:

1. Assign a logical interface of the C-VLAN interface to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

2. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set** flexible-vlan-tagging

3. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set** encapsulation **extended-vlan-bridge**

5. Configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

6. Configure the system to remove the existing C-VLAN tag and replace it with the S-VLAN tag when packets ingress on the C-VLAN interface and are forwarded to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** input-vlan-map swap

7. Configure the system to remove the existing S-VLAN tag and replace it with the C-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** output-vlan-map swap

8. To configure an S-VLAN and associate it with the appropriate C-VLAN interface:

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name***

For example, the following configuration on C-VLAN interface xe-0/0/1.200 enables Q-in-Q tunneling, enables xe-0/0/1 to accept untagged packets, and maps incoming packets from C-VLAN 150 to logical interface 200, which is a member of S-VLAN 200. Also, when packets egress from C-VLAN interface xe-0/0/1 and travel to the S-VLAN interface, the C-VLAN tag of 150 is removed and replaced with the S-VLAN tag of 200.

When packets travel from the S-VLAN interface to the C-VLAN interface, the S-VLAN tag of 200 is removed and replaced with the C-VLAN tag of 150.

```
set vlans v200 interface xe-0/0/1.200
set interfaces xe-0/0/1 flexible-vlan-tagging
set interfaces xe-0/0/1 native-vlan-id 150
set interfaces xe-0/0/1 encapsulation extended-vlan-bridge
set interfaces xe-0/0/1 unit 200 vlan-id 200
set interfaces xe-0/0/1 unit 200 output-vlan-map swap
set interfaces xe-0/0/1 unit 200 input-vlan-map swap
```

See Also

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using All-in-One Bundling on page 565
- Configuring Q-in-Q Tunneling Using Many-to-Many Bundling on page 567

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554

Configuring Q-in-Q Tunneling on Security Devices

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.

NOTE: Q-in-Q VLAN tagging is supported only on SRX340, SRX345, SRX550M, and SRX1500 devices.

NOTE: VLAN translation is supported on SRX300 and SRX320 devices and these devices do not support Q-in-Q tunneling.

Q-in-Q tunneling prepends a service VLAN tag to all customer's 802.1Q VLAN tags. The Juniper Networks Junos OS implementation of Q-in-Q tunneling supports the IEEE 802.1ad standard.

NOTE: This task uses a Junos OS release that supports the Enhanced Layer 2 Software (ELS) configuration style.

With releases earlier than Junos OS Release 15.1X49-D80, you cannot create a regular VLAN on an interface if you have created an S-VLAN or C-VLAN on that interface for Q-in-Q tunneling. This means that you cannot create an integrated routing and bridging (IRB) interface on that interface because regular VLANs are a required part of and IRB

configuration. With Junos OS Release 15.1X49-D80, you can create a regular VLAN on a trunk interface that has an S-VLAN, which means that you can also create an IRB interface on the trunk. In this case, the regular VLAN and S-VLAN on the same trunk interface cannot share the same VLAN ID. Junos OS Release 15.1X49-D80, does not allow you to create a regular VLAN on an access interface that has a C-VLAN.

Before setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs on the neighboring devices. See *Example: Configuring VLANs on Security Devices (J-Web Procedure)*.

Using the Different Mapping Methods

Once you have created the required VLANs on the neighboring devices, configure Q-in-Q tunneling using one of the three methods to map customer VLANs (C-VLANs) to service-provider-defined service VLANs (S-VLANs):

- All-in-one bundling maps all packets from all C-VLAN interfaces to an S-VLAN.
- Use many-to-many bundling when you want a subset of the C-VLANs on the access device to be part of multiple S-VLANs.
- Use specific interface mapping when you want to assign an S-VLAN to a specific C-VLAN on an interface.

Configuring All-in-One Bundling

You can configure Q-in-Q tunneling using the all-in-one bundling method, which forwards all packets entering a C-VLAN interface to an S-VLAN. (Packets are forwarded to the S-VLAN regardless of whether they are tagged or untagged before they enter.) Using this approach saves you the effort of specifying a specific mapping for each C-VLAN.

First configure the S-VLAN and its interface:

1. Enable the interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@host#` **flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@host#` **encapsulation extended-vlan-bridge**

3. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]

```
user@host# native-vlan-id vlan-id
```

4. Bind the logical interface (unit) of the interface to the automatically-created VLAN ID for the S-VLAN:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# vlan-id number
user@host# family ethernet-switching vlan members vlain-id
```

For example, the following configuration enables Q-in-Q tunneling on interface ge-0/0/7, enables ge-0/0/7 to accept untagged packets, and binds the VLAN ID of S-VLAN VL-S91 to a logical interface of ge-0/0/7.

```
set interfaces ge-0/0/7 flexible-vlan-tagging
set interfaces ge-0/0/7 native-vlan-id 91
set interfaces ge-0/0/7 encapsulation extended-vlan-bridge
set interfaces ge-0/0/7 unit 91 vlan-id 91
set interfaces ge-0/0/7 unit 91 family ethernet-switching vlan members VL-S91
```

Now configure all-in-one bundling on a C-VLAN interface:

1. Enable the interface to transmit packets with 802.1Q VLAN tags:

```
[edit interfaces interface-name]
user@host# flexible-vlan-tagging
```

2. Enable extended VLAN bridge encapsulation on the interface:

```
[edit interfaces interface-name]
user@host# encapsulation extended-vlan-bridge
```

3. Enable the C-VLAN interface to send and receive untagged packets:

```
[edit interfaces interface-name]
user@host# native-vlan-id vlan-id
```

4. Configure a logical interface to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# vlan-id-list vlan-id-numbers
```

NOTE: On some SRX Series devices, you can apply no more than eight VLAN identifier lists to a physical interface.

5. Configure the system to add an S-VLAN tag (outer tag) as packets travel from a C-VLAN interface to the S-VLAN:

```
[edit interfaces interface-name unit logical-unit-number]
user@host# input-vlan-map push
```

6. Configure the system to remove the S-VLAN tag when packets are forwarded (internally) from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@host#` **output-vlan-map pop**
   `user@host#` **family ethernet-switching vlan members *vlan-id***

7. Configure S-VLAN and vlan id binding:

   [edit vlans *vlan-name*]
   `user@host#` **vlan-id *vlan-id-numbers***

For example, the following configuration makes ge-0/0/4 a member of S-VLAN VL-S91, enables Q-in-Q tunneling, maps packets from C-VLANs to S-VLAN VL-S91, and enables ge-0/0/4 to accept untagged packets. If a packet originates in C-VLAN and needs to be sent across the S-VLAN, a tag with VLAN ID 91 is added to the packet. When a packet is forwarded (internally) from the S-VLAN interface to interface ge-0/0/4, the tag with VLAN ID 91 is removed.

```
set interfaces ge-0/0/4 flexible-vlan-tagging
set interfaces ge-0/0/4 native-vlan-id 50
set interfaces ge-0/0/4 encapsulation extended-vlan-bridge
set interfaces ge-0/0/4 unit 50 vlan-id-list 30-70
set interfaces ge-0/0/4 unit 50 input-vlan-map push
set interfaces ge-0/0/4 unit 50 output-vlan-map pop
set interfaces ge-0/0/4 unit 50 family ethernet-switching vlan members VL-S91
set vlans VL-S91 vlan-id 91
```

Configuring Many-to-Many Bundling

You can configure Q-in-Q tunneling using the many-to-many bundling method, which maps packets from multiple C-VLANs to multiple S-VLANs. This method is convenient for mapping a range of C-VLANs without having to specify each one individually. (You can also use this method to configure only one C-VLAN to be mapped to an S-VLAN.)

First configure the S-VLANs and assign them to an interface:

1. Enable the physical interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@host#` **flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@host#` **encapsulation extended-vlan-bridge**

3. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@host#` **native-vlan-id *vlan-id***

4. Bind one of the logical units of the interface to the VLAN ID for one of the S-VLANs.

   [edit interfaces *interface-name* unit *logical-unit-number*]

```
user@host# vlan-id number
user@host# family ethernet-switching vlan members vlan-id
```

5. Repeat Step 4 to bind the automatically-created VLAN IDs for the other S-VLANs to the other logical units of the interface:

For example, the following configuration creates S-VLANs VL-S10 and VL-S30 and associates them with interface ge-0/0/7. It also enables Q-in-Q tunneling, enables ge-0/0/7 to accept untagged packets, and maps incoming C-VLAN packets to S-VLANs VL-S10 and VL-S30.

```
set interfaces ge-0/0/7 flexible-vlan-tagging
set interfaces ge-0/0/7 native-vlan-id 10
set interfaces ge-0/0/7 encapsulation extended-vlan-bridge
set interfaces ge-0/0/7 unit 10 vlan-id 10
set interfaces ge-0/0/7 unit 10 family ethernet-switching vlan members VL-S10
set interfaces ge-0/0/7 unit 30 vlan-id 30
set interfaces ge-0/0/7 unit 30 family ethernet-switching vlan members VL-S30
```

To configure the many-to-many bundling method on a C-VLAN interface, perform the following steps for each customer:

1. Enable the interface to transmit packets with 802.1Q VLAN tags:

   ```
   [edit interfaces interface-name]
   user@host# flexible-vlan-tagging
   ```

2. Enable extended VLAN bridge encapsulation on the interface:

   ```
   [edit interfaces interface-name]
   user@host# encapsulation extended-vlan-bridge
   ```

3. Enable the C-VLAN interface to send and receive untagged packets:

   ```
   [edit interfaces interface-name]
   user@host# native-vlan-id vlan-id
   ```

4. For each physical interface, configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

   ```
   [edit interfaces interface-name unit logical-unit-number]
   user@host# vlan-id-list vlan-id-numbers
   ```

   To configure only one C-VLAN to be mapped to an S-VLAN, specify only one VLAN ID after vlan-id-list.

   NOTE: On some SRX Series devices you can apply no more than eight VLAN identifier list to a physical interface.

5. For each physical interface, configure the system to add an S-VLAN tag (outer tag) as packets travel from the C-VLAN interface to the S-VLAN:

[edit interfaces *interface-name* unit *logical-unit-number*]
user@host# input-vlan-map push

6. For each physical interface, configure the system to remove the S-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   user@host# output-vlan-map pop
   user@host# family ethernet-switching vlan members *vlan-id*

7. Configure S-VLAN and vlan id binding:

   [edit vlans *vlan-name*]
   user@host# vlan-id *vlan-id-numbers*

For example, the following configuration makes ge-0/0/1 a member of S-VLAN VL-S10, enables Q-in-Q tunneling, and maps packets from C-VLANs 10 through 20 to S-VLAN VL-S10. The configuration for customer 2 makes ge-0/0/2 a member of S-VLAN VL-S30, enables Q-in-Q tunneling, and maps packets from C-VLANs 30 through 40, 50 through 60, and 70 through 80 to S-VLAN VL-S30. Both interfaces are configured to accept untagged packets.

If a packet originates in C-VLAN 10 and needs to be sent over the S-VLAN, a tag with a VLAN 10 is added to the packet. If a packet is forwarded internally from the S-VLAN interface to ge-0/0/1, the tag with VLAN 10 is removed. The same principles apply to the C-VLANs configured on interface ge-0/0/2.

NOTE: Notice that you can use the same tag value for an S-VLAN and C-VLAN. For example, the configuration for customer 1 maps C-VLAN 10 to S-VLAN VL-S10. Because C-VLAN and S-VLAN tags use separate name spaces, this configuration is allowed.

Configuration for customer 1:

```
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 10 vlan-id-list 10-20
set interfaces ge-0/0/1 native-vlan-id 15
set interfaces ge-0/0/1 unit 10 input-vlan-map push
set interfaces ge-0/0/1 unit 10 output-vlan-map pop
set interfaces ge-0/0/1 unit 10 family ethernet-switching vlan members VL-S10
set vlans VL-S10 vlan-id 10
```

Configuration for customer 2:

```
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 encapsulation extended-vlan-bridge
set interfaces ge-0/0/2 unit 30 vlan-id-list 30-40
set interfaces ge-0/0/2 unit 30 vlan-id-list 50-60
set interfaces ge-0/0/2 unit 30 vlan-id-list 70-80
set interfaces ge-0/0/2 native-vlan-id 75
set interfaces ge-0/0/2 unit 30 input-vlan-map push
```

```
set interfaces ge-0/0/2 unit 30 output-vlan-map pop
set interfaces ge-0/0/2 unit 30 family ethernet-switching vlan members VL-S30
set vlans VL-S30 vlan-id 30
```

Configuring a Specific Interface Mapping with VLAN ID Translation Option

You can configure Q-in-Q tunneling by mapping packets from a specified C-VLAN to a specified S-VLAN. In addition, you can configure the system to replace a C-VLAN tag with an S-VLAN tag or replace an S-VLAN tag with a C-VLAN tag (instead of double tagging). This is called VLAN translation or VLAN rewriting. VLAN translation is particularly useful if a service provider's Layer 2 network that connects to customer sites does not support double tagged packets.

When you use VLAN translation, both ends of the link normally must be able to swap the tags appropriately. That is, both ends of the link must be configured to swap the C-VLAN tag for the S-VLAN tag and swap the S-VLAN tag for the C-VLAN tag so that traffic in both directions is tagged appropriately while in transit and after arrival.

First configure the S-VLAN and its interface:

1. Enable the interface to transmit packets with 802.1Q VLAN tags:

   ```
   [edit interfaces interface-name]
   user@host# flexible-vlan-tagging
   ```

2. Enable the S-VLAN interface to send and receive untagged packets:

   ```
   [edit interfaces interface-name]
   user@host# native-vlan-id vlan-id
   ```

3. Enable extended VLAN bridge encapsulation on the interface:

   ```
   [edit interfaces interface-name]
   user@host# encapsulation extended-vlan-bridge
   ```

4. Bind the logical interface (unit) of the interface that you specified earlier to the VLAN ID for the S-VLAN:

   ```
   [edit interfaces interface-name unit logical-unit-number]
   user@host# vlan-id number
   user@host# family ethernet-switching vlan members vlan-id
   ```

For example, the following configuration enables Q-in-Q tunneling on interface ge-0/0/0, enables ge-0/0/0 to accept untagged packets, and binds a logical interface of ge-0/0/0 to the VLAN ID of S-VLAN VL-S200.

```
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 native-vlan-id 10
set interfaces ge-0/0/0 encapsulation extended-vlan-bridge
set interfaces ge-0/0/0 unit 200 vlan-id 200
set interfaces ge-0/0/0 unit 200 family ethernet-switching vlan members VL-S200
```

Now configure a specific interface mapping with optional VLAN ID translation on the C-VLAN interface:

1. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@host#` **flexible-vlan-tagging**

2. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@host#` **native-vlan-id *vlan-id***

3. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@host#` **encapsulation extended-vlan-bridge**

4. Configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@host#` **vlan-id *number***

5. Configure the system to remove the existing C-VLAN tag and replace it with the S-VLAN tag when packets enter the C-VLAN interface and are forwarded to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@host#` **input-vlan-map swap**

6. Configure the system to remove the existing S-VLAN tag and replace it with the C-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@host#` **output-vlan-map swap**

7. To configure an S-VLAN and associate it with the appropriate C-VLAN interface:

   [edit vlans *vlan-name*]
   `user@host#` **interface *interface-name***

8. Configure S-VLAN and vlan id binding:

   [edit vlans *vlan-name*]
   `user@host#` **vlan-id *vlan-id-numbers***

For example, the following configuration on C-VLAN interface ge-0/0/1 enables Q-in-Q tunneling, enables ge-0/0/1 to accept untagged packets, and maps incoming packets from C-VLAN 150 to logical interface 200, which is a member of S-VLAN VL-S200. Also, when packets exit from C-VLAN interface ge-0/0/1 and travel to the S-VLAN interface, the C-VLAN tag of 150 is removed and replaced with the S-VLAN tag of 200. When

packets travel from the S-VLAN interface to the C-VLAN interface, the S-VLAN tag of 200 is removed and replaced with the C-VLAN tag of 150.

```
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 native-vlan-id 10
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 200 vlan-id 150
set interfaces ge-0/0/1 unit 200 family ethernet-switching vlan members VL-S200
set interfaces ge-0/0/1 unit 200 output-vlan-map swap
set interfaces ge-0/0/1 unit 200 input-vlan-map swap
Set vlans VL-S200 vlan-id 200
```

Configuring Q-in-Q Tunneling on QFX Series Switches

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.

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 Switches" on page 93.

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-ids-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 554
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598
- *Troubleshooting Q-in-Q and VLAN Translation Configuration*
- *mtu*

Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches 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 on EX Series Switches with ELS Support (CLI Procedure)" on page 583. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

Q-in-Q tunneling allows service providers on Ethernet access networks to segregate or bundle customer traffic into different VLANs by adding another layer of 802.1Q tags. You can configure Q-in-Q tunneling on EX Series switches.

NOTE: You cannot configure 802.1X user authentication on interfaces that have been enabled for Q-in-Q tunneling.

Before you begin configuring Q-in-Q tunneling, make sure you set up your VLANs. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98 or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

To configure Q-in-Q tunneling:

1. Enable Q-in-Q tunneling on the S-VLAN:

   [edit vlans]
   `user@switch#` **set *s-vlan-name*** dot1q-tunneling

2. Set the allowed C-VLANs on the S-VLAN (optional). Here, the C-VLANs are identified by VLAN range:

   [edit vlans]
   `user@switch#` **set *s-vlan-name* dot1q-tunneling** customer-vlans ***range***

3. Change the global Ethertype value (optional):

   [edit]

```
user@switch# set ethernet-switching-options dot1q-tunneling ether-type ether-type-value
```

4. Disable MAC address learning on the S-VLAN (optional):

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

Related Documentation

- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- Verifying That Q-in-Q Tunneling Is Working on Switches on page 607
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554

Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

Q-in-Q tunneling enables service providers on Ethernet access networks to segregate or bundle customer traffic into different VLANs by adding another layer of 802.1Q tags. You can configure Q-in-Q tunneling on EX Series switches.

NOTE: You cannot configure 802.1X user authentication on interfaces that have been enabled for Q-in-Q tunneling.

When Q-in-Q tunneling is configured on EX Series switches, trunk interfaces are assumed to be part of the service-provider network and access interfaces are assumed to be part of the customer network. Therefore, this topic also refers to trunk interfaces as service-provider VLAN (S-VLAN) interfaces (network-to-network interfaces [NNI]), and to access interfaces as customer VLAN (C-VLAN) interfaces (user-network interfaces [UNI]).

Before you begin configuring Q-in-Q tunneling, make sure you set up your VLANs. See "Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure)" on page 102 or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

Configure Q-in-Q tunneling by using one of the following methods to map C-VLANs to S-VLANs:

- Configuring All-in-One Bundling on page 584
- Configuring Many-to-Many Bundling on page 585
- Configuring a Specific Interface Mapping with VLAN Rewrite Option on page 588

Configuring All-in-One Bundling

You can configure Q-in-Q tunneling by using the all-in-one bundling method, which maps packets from all C-VLAN interfaces on a switch to an S-VLAN.

To configure the all-in-one bundling method on a C-VLAN interface:

1. Enable the transmission of packets with no or a single 802.1Q VLAN tag:

   [edit interfaces *interface-name*]
   `user@switch#` **set** flexible-vlan-tagging

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set** encapsulation **extended-vlan-bridge**

3. Map packets from all C-VLANs to a logical interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** vlan-id-list ***vlan-id-numbers***

   CAUTION: You can apply no more than eight VLAN identifier lists to a physical interface.

4. Enable a C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set** native-vlan-id ***vlan-id***

   When specifying a native VLAN ID on a C-VLAN physical interface, the value must be included in the VLAN ID list specified on the C-VLAN logical interface in step 3.

5. Specify that packets traveling from a C-VLAN interface to an S-VLAN interface are tagged with the VLAN ID of the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** input-vlan-map push

6. Specify that the 802.1Q S-VLAN tag is removed as packets exit an S-VLAN interface.

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** output-vlan-map pop

7. Configure a name for the S-VLAN, and associate the logical interface configured in step 3 with the S-VLAN:

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.logical-unit-number***

The following configuration on the C-VLAN interface ge-0/0/1 enables Q-in-Q tunneling and maps packets from C-VLANs 100 through 200 to logical interface 10, which is in turn associated with S-VLAN v10. In this sample configuration, a packet originated in C-VLAN 100 includes a tag with the VLAN ID 100. When this packet travels from the interface ge-0/0/1 to the S-VLAN interface, a tag with VLAN ID 10 is added to it. As the packet exits the S-VLAN interface, the tag with the VLAN ID 10 is removed. .

```
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 10 vlan-id-list 100-200
set interfaces ge-0/0/1 native-vlan-id 150
set interfaces ge-0/0/1 unit 10 input-vlan-map push
set interfaces ge-0/0/1 unit 10 output-vlan-map pop
set vlans v10 interface ge-0/0/1.10
```

To configure the all-in-one bundling method on an S-VLAN interface:

1. Enable the transmission of packets with no, one, or two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from the logical interface specified in the C-VLAN interface configuration to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** vlan-id ***number***

4. Associate the S-VLAN interface with the S-VLAN that was configured in the C-VLAN interface procedure:

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.logical-unit-number***

For example, the following configuration on the S-VLAN interface ge-1/1/1 enables Q-in-Q tunneling and maps packets with a VLAN ID tag of 10 to logical interface 10, which is in turn associated with S-VLAN v10. .

```
set interfaces ge-1/1/1 flexible-vlan-tagging
set interfaces ge-1/1/1 encapsulation extended-vlan-bridge
set interfaces ge-1/1/1 unit 10 vlan-id 10
set vlans v10 interface ge-1/1/1.10
```

Configuring Many-to-Many Bundling

You can configure Q-in-Q tunneling by using the many-to-many bundling method, which maps packets from multiple C-VLANs to multiple S-VLANs.

To configure the many-to-many bundling method on a C-VLAN interface:

1. Enable the transmission of packets with no or a single 802.1Q VLAN tag:

   [edit interfaces *interface-name*]
   `user@switch#` **set** flexible-vlan-tagging

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set** encapsulation **extended-vlan-bridge**

3. Map packets from specified C-VLANs to a logical interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** vlan-id-list ***vlan-id-numbers***

4. Enable a C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set** native-vlan-id ***vlan-id***

   When specifying a native VLAN ID on a C-VLAN physical interface, the value must be included in the VLAN ID list specified on the C-VLAN logical interface in step 3.

5. Specify that packets traveling from a C-VLAN interface to an S-VLAN interface are tagged with the VLAN ID of the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** input-vlan-map push

6. Specify that the 802.1Q S-VLAN tag is removed as packets exit an S-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** output-vlan-map pop

7. Configure a name for an S-VLAN, and associate the logical interface configured in step 3 with the S-VLAN:

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.logical-unit-number***

The following configuration on the C-VLAN interface ge-0/0/1 for customer 1 enables Q-in-Q tunneling and maps packets from C-VLANs 100 through 120 to logical interface 10, which is in turn associated with S-VLAN v10.

The configuration on the C-VLAN interface ge-0/0/2 for customer 2 enables Q-in-Q tunneling and maps packets from C- VLANs 30 through 40, 50 through 60, and 70 through 80 to logical interface 30, which is in turn associated with S- VLAN v30.

In this sample configuration, a packet originated in C-VLAN 100 includes a tag with the VLAN ID 100. When this packet travels from the interface ge-0/0/1 to the S-VLAN

interface, a tag with a VLAN ID of 10 is added to it. As the packet exits the S-VLAN interface, the tag with the VLAN ID of 10 is removed.

Customer 1

```
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 10 vlan-id-list 100-120
set interfaces ge-0/0/1 native-vlan-id 100
set interfaces ge-0/0/1 unit 10 input-vlan-map push
set interfaces ge-0/0/1 unit 10 output-vlan-map pop
set vlans v10 interface ge-0/0/1.10
```

Customer 2

```
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 encapsulation extended-vlan-bridge
set interfaces ge-0/0/2 unit 30 vlan-id-list 30-40
set interfaces ge-0/0/2 unit 30 vlan-id-list 50-60
set interfaces ge-0/0/2 unit 30 vlan-id-list 70-80
set interfaces ge-0/0/2 native-vlan-id 30
set interfaces ge-0/0/2 unit 30 input-vlan-map push
set interfaces ge-0/0/2 unit 30 output-vlan-map pop
set vlans v30 interface ge-0/0/2.30
```

To configure the many-to-many bundling method on an S-VLAN interface:

1. Enable the transmission of packets with no, one, or two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from each logical interface specified in the C-VLAN interface configuration to an S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set**vlan-id ***number***

4. Enable an S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

   When specifying a native VLAN ID on an S-VLAN physical interface, the value must match an S-VLAN ID specified on the S-VLAN logical interface in step 3.

5. Associate the S-VLAN interface with the S-VLANs that were configured in the C-VLAN interface procedure:

   [edit vlans *vlan-name*]

```
user@switch# set interface interface-name.logical-unit-number
```

For example, the following configuration on the S-VLAN interface ge-1/1/1 enables Q-in-Q tunneling and maps incoming C-VLAN packets to logical interfaces 10 and 30, which are in turn associated with S-VLANs v10 and v30, respectively.

```
set interfaces ge-1/1/1 flexible-vlan-tagging
set interfaces ge-1/1/1 encapsulation extended-vlan-bridge
set interfaces ge-1/1/1 unit 10 vlan-id 10
set interfaces ge-1/1/1 unit 30 vlan-id 30
set interfaces ge-1/1/1 native-vlan-id 10
set vlans v10 interface ge-1/1/1.10
set vlans v30 interface ge-1/1/1.30
```

Configuring a Specific Interface Mapping with VLAN Rewrite Option

You can configure Q-in-Q tunneling by mapping packets from a specified C-VLAN to a specified S-VLAN. In addition, while the packets are transmitted to and from the S-VLAN, you can specify that the 802.1Q C-VLAN tag be removed and replaced with the S-VLAN tag or vice versa.

To configure a specific interface mapping with VLAN rewriting on the C-VLAN interface:

1. Enable the transmission of packets with no or one 802.1Q VLAN tag:

   ```
   [edit interfaces interface-name]
   user@switch# set flexible-vlan-tagging
   ```

2. Enable extended VLAN bridge encapsulation:

   ```
   [edit interfaces interface-name]
   user@switch# set encapsulation extended-vlan-bridge
   ```

3. Map packets from a specified C-VLAN to a logical interface:

   ```
   [edit interfaces interface-name unit logical-unit-number]
   user@switch# set vlan-id number
   ```

4. Enable the C-VLAN interface to send and receive untagged packets:

   ```
   [edit interfaces interface-name]
   user@switch# set native-vlan-id vlan-id
   ```

   When specifying a native VLAN ID on a C-VLAN physical interface, the value must match the VLAN ID specified on the C-VLAN logical interface in step 3.

5. Specify that the existing 802.1Q C-VLAN tag is removed from packets traveling from a C-VLAN interface to an S-VLAN interface and replaced with the 802.1Q S-VLAN tag:

   ```
   [edit interfaces interface-name unit logical-unit-number]
   user@switch# set input-vlan-map swap
   ```

6. Specify that the existing 802.1Q S-VLAN tag is removed from packets traveling from an S-VLAN interface to a C-VLAN interface and replaced with the 802.1Q C-VLAN tag:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** output-vlan-map swap

7. Configure a name for the S-VLAN, and associate the logical interface configured in step 3 with the S-VLAN:

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.logical-unit-number***

For example, the following configuration on the C-VLAN interface ge-0/0/1 enables Q-in-Q tunneling and maps incoming packets from C-VLAN 150 to logical interface 200, which is in turn associated with VLAN v200. Also, as packets travel from the C-VLAN interface ge-0/0/1 to an S-VLAN interface, the C-VLAN tag 150 is removed and replaced with the S-VLAN tag 200. As packets travel from an S-VLAN interface to C-VLAN interface ge-0/0/1, the S-VLAN tag 200 is removed and replaced with the C-VLAN tag of 150.

```
set interfaces ge-0/0/1 flexible-vlan-tagging
set interfaces ge-0/0/1 encapsulation extended-vlan-bridge
set interfaces ge-0/0/1 unit 200 vlan-id 150
set interfaces ge-0/0/1 native-vlan-id 150
set interfaces ge-0/0/1 unit 200 input-vlan-map swap
set interfaces ge-0/0/1 unit 200 output-vlan-map swap
set vlans v200 interface ge-0/0/1.200
```

To configure a specific interface mapping with VLAN rewriting on the S-VLAN interface:

1. Enable the transmission of packets with no, one, or two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

2. Enable extended VLAN bridge encapsulation:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

3. Map packets from the logical interface specified in the C-VLAN interface configuration to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** vlan-id ***number***

4. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

When specifying a native VLAN ID on an S-VLAN physical interface, the value must match the VLAN ID specified on the S-VLAN logical interface in step 3.

5. Associate the S-VLAN interface with the S-VLAN that was configured in the C-VLAN interface procedure: :

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.logical-unit-number***

For example, the following configuration on the S-VLAN interface ge-1/1/1 enables Q-in-Q tunneling and maps packets with VLAN ID 200 to logical interface 200, which is in turn associated with S-VLAN v200.

```
set interfaces ge-1/1/1 flexible-vlan-tagging
set interfaces ge-1/1/1 encapsulation extended-vlan-bridge
set interfaces ge-1/1/1 unit 200 vlan-id 200
set interfaces ge-1/1/1 native-vlan-id 200
set vlans v200 interface ge-1/1/1.200
```

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554

Configuring Q-in-Q Tunneling Using All-in-One Bundling

You can configure Q-in-Q tunneling using the all-in-one bundling method, which forwards all packets that ingress on a C-VLAN interface to an S-VLAN. (Packets are forwarded to the S-VLAN regardless of whether they are tagged or untagged prior to ingress.) Using this approach saves you the effort of specifying a specific mapping for each C-VLAN.

First configure the S-VLAN and its interface:

1. Assign a logical interface (unit) to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **interface *interface-name.unit-number***

   NOTE: Do not use logical interface unit 0. You must later bind a VLAN tag ID to the unit you specify in this step, and you cannot bind a VLAN tag ID to unit 0. Also note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Enable the interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]

```
user@switch# set encapsulation extended-vlan-bridge
```

NOTE: If you configure an enterprise-style configuration such as PVLAN on the same physical interface on which you are configuring Q-in-Q tunneling, use set encapsulation flexible-ethernet-services in step 3. See *Understanding Flexible Ethernet Services Encapsulation on Switches*.

4. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   user@switch# set native-vlan-id *vlan-id*

5. Bind the logical interface (unit) of the interface that you specified in step 1 to the automatically created VLAN ID for the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   user@switch# set vlan-id *number*

NOTE: If you configured flexible-ethernet-services in step 3, configure vlan-bridge encapsulation on the logical interface:

[edit interfaces *interface-name* unit *logical-unit-number*]
user@switch# set encapsulation vlan-bridge

For example, the following configuration makes xe-0/0/0.10 a member of VLAN 10, enables Q-in-Q tunneling on interface xe-0/0/0, enables xe-0/0/0 to accept untagged packets, and binds the VLAN ID of S-VLAN v10 to a logical interface of xe-0/0/0.

```
set vlans v10 interface xe-0/0/0.10
set interfaces xe-0/0/0 flexible-vlan-tagging
set interfaces xe-0/0/0 native-vlan-id 10
set interfaces xe-0/0/0 encapsulation extended-vlan-bridge
set interfaces xe-0/0/0 unit 10 vlan-id 10
```

Now configure all-in-one bundling on a C-VLAN interface:

1. Assign a logical interface (unit) of the C-VLAN interface to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   user@switch# set interface *interface-name.unit-number*

2. Enable the interface to transmit packets with 802.1Q VLAN tags :

   [edit interfaces *interface-name*]
   user@switch# set flexible-vlan-tagging

3. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]

```
user@switch# set encapsulation extended-vlan-bridge
```

4. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   user@switch# set native-vlan-id *vlan-id*

5. Configure a logical interface to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   user@switch# set vlan-id-list *vlan-id-numbers*

   CAUTION: You can apply no more than eight VLAN identifier lists to a physical interface. This limitation does not apply to QFX10000 switches.

6. Configure the system to add an S-VLAN tag (outer tag) as packets travel from a C-VLAN interface to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   user@switch# set input-vlan-map push

   NOTE: You can configure vlan-id on input-vlan-map, but doing so is optional.

7. Configure the system to remove the S-VLAN tag when packets are forwarded (internally) from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   user@switch# set output-vlan-map pop

For example, the following configuration makes xe-0/0/1.10 a member of S-VLAN v10, enables Q-in-Q tunneling, maps packets from C-VLANs 100 through 200 to S-VLAN 10, and enables xe-0/0/1 to accept untagged packets. If a packet originates in C-VLAN 100 and needs to be sent across the S-VLAN, a tag with VLAN ID 10 is added to the packet. When a packet is forwarded (internally) from the S-VLAN interface to interface xe-0/0/1, the tag with VLAN ID 10 is removed.

```
set vlans v10 interface xe-0/0/1.10
set interfaces xe-0/0/1 flexible-vlan-tagging
set interfaces xe-0/0/1 encapsulation extended-vlan-bridge
set interfaces xe-0/0/1 unit 10 vlan-id-list 100-200
set interfaces xe-0/0/1 native-vlan-id 150
set interfaces xe-0/0/1 unit 10 input-vlan-map push
set interfaces xe-0/0/1 unit 10 output-vlan-map pop
```

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using Many-to-Many Bundling on page 567
- Configuring a Specific Interface Mapping with VLAN ID Translation Option on page 570

Configuring Q-in-Q Tunneling Using Many-to-Many Bundling

You can configure Q-in-Q tunneling using the many-to-many bundling method, which maps packets from multiple C-VLANs to multiple S-VLANs. This method is convenient for mapping a range of C-VLANs without having to specify each one individually. (You can also use this method to configure only one C-VLAN to be mapped to an S-VLAN.)

First configure the S-VLANs and assign them to an interface:

1. Assign a logical interface (unit) to be a member of one of the S-VLANs. Do not use logical interface unit 0.

   [edit vlans *vlan-name*]
   user@switch# set interface *interface-name.unit-number*

   NOTE: Note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Repeat step 1 for the other S-VLANs.

3. Enable the physical interface to transmit packets with two 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   user@switch# set flexible-vlan-tagging

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   user@switch# set encapsulation extended-vlan-bridge

5. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   user@switch# set native-vlan-id *vlan-id*

6. Bind one of the logical units of the interface to the VLAN ID for one of the S-VLANs.

   [edit interfaces *interface-name* unit *logical-unit-number*]
   user@switch# set vlan-id *number*

7. Repeat step 6 to bind the automatically-created VLAN IDs for the other S-VLANs to the other logical units of the interface:

For example, the following configuration creates S-VLANs v10 and v30 and associates them with interface xe-0/0/0.10, enables Q-in-Q tunneling, enables xe-0/0/0 to accept untagged packets, and maps incoming C-VLAN packets to S-VLANs v10 and v30.

```
set vlans v10 interface xe-0/0/0.10
set vlans v30 interface xe-0/0/0.10
set interfaces xe-0/0/0 flexible-vlan-tagging
set interfaces xe-0/0/0 native-vlan-id 10
set interfaces xe-0/0/0 encapsulation extended-vlan-bridge
set interfaces xe-0/0/0 unit 10 vlan-id 10
set interfaces xe-0/0/0 unit 30 vlan-id 30
```

To configure the many-to-many bundling method on a C-VLAN interface, perform the following steps for each customer:

1. Assign a logical interface (unit) of one C-VLAN interface to be a member of one S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

2. Repeat step 1 to assign another C-VLAN interface (physical interface) to be a member of another S-VLAN.

3. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set** flexible-vlan-tagging

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` encapsulation **extended-vlan-bridge**

5. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

6. For each physical interface, configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the list of VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id-list *vlan-id-numbers***

   To configure only one C-VLAN to be mapped to an S-VLAN, specify only one VLAN ID after *vlan-id-list*.

CAUTION: You can apply no more than eight VLAN identifier lists to a physical interface. This limitation does not apply to QFX10000 switches.

7. For each physical interface, configure the system to add an S-VLAN tag (outer tag) as packets travel from the C-VLAN interface to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch# set input-vlan-map push`

8. For each physical interface, configure the system to remove the S-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch# set output-vlan-map pop`

For example, the following configuration makes xe-0/0/1.10 a member of S-VLAN v10, enables Q-in-Q tunneling, and maps packets from C-VLANs 10 through 20 to S-VLAN 10. The configuration for customer 2 makes xe-0/0/2.30 a member of S-VLAN v30, enables Q-in-Q tunneling, and maps packets from C-VLANs 30 through 40, 50 through 60, and 70 through 80 to S-VLAN 30. Both interfaces are configured to accept untagged packets.

If a packet originates in C-VLAN 10 and needs to be sent over the S-VLAN, a tag with a VLAN ID 10 is added to the packet. If a packet is forwarded internally from the S-VLAN interface to xe-0/0/1.10, the tag with VLAN ID 10 is removed. The same principles apply to the C-VLANs configured on interface xe-0/0/2.

NOTE: Notice that you can use the same tag value for an S-VLAN and C-VLAN. For example, the configuration for customer 1 maps C-VLAN ID 10 to S-VLAN ID 10. C-VLAN and S-VLAN tags use separate name spaces, so this configuration is allowed.

Configuration for customer 1:

```
set vlans v10 interface xe-0/0/1.10
set interfaces xe-0/0/1 flexible-vlan-tagging
set interfaces xe-0/0/1 encapsulation extended-vlan-bridge
set interfaces xe-0/0/1 unit 10 vlan-id-list 10-20
set interfaces xe-0/0/1 native-vlan-id 15
set interfaces xe-0/0/1 unit 10 input-vlan-map push
set interfaces xe-0/0/1 unit 10 output-vlan-map pop
```

Configuration for customer 2:

```
set vlans v30 interface xe-0/0/2.30
set interfaces xe-0/0/2 flexible-vlan-tagging
set interfaces xe-0/0/2 encapsulation extended-vlan-bridge
set interfaces xe-0/0/2 unit 30 vlan-id-list 30-40
set interfaces xe-0/0/2 unit 30 vlan-id-list 50-60
set interfaces xe-0/0/2 unit 30 vlan-id-list 70-80
set interfaces xe-0/0/2 native-vlan-id 75
set interfaces xe-0/0/2 unit 30 input-vlan-map push
set interfaces xe-0/0/2 unit 30 output-vlan-map pop
```

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using All-in-One Bundling on page 565
- Configuring a Specific Interface Mapping with VLAN ID Translation Option on page 570

Configuring a Specific Interface Mapping with VLAN ID Translation Option

You can configure Q-in-Q tunneling by mapping packets from a specified C-VLAN to a specified S-VLAN. In addition, you can configure the system to replace a C-VLAN tag with an S-VLAN tag or replace an S-VLAN tag with a C-VLAN tag (instead of double tagging). This is call VLAN translation or VLAN rewriting. VLAN translation is particularly useful if a service provider's Layer 2 network that connects a customer's sites does not support double tagged packets.

When you use VLAN translation, both ends of the link normally must be able to swap the tags appropriately. That is, both ends of the link must be configured to swap the C-VLAN tag for the S-VLAN tag and swap the S-VLAN tag for the C-VLAN tag so that traffic in both directions is tagged appropriately while in transit and after arrival.

First configure the S-VLAN and its interface:

1. Assign a logical interface to be a member of the S-VLAN. Do not use unit 0.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

   NOTE: Note that you do not create a VLAN ID for the S-VLAN. The ID is created automatically for the appropriate logical interface.

2. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set flexible-vlan-tagging**

3. Enable the S-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set encapsulation extended-vlan-bridge**

5. Bind the logical interface (unit) of the interface that you specified earlier to the VLAN ID for the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

For example, the following configuration creates S-VLAN v200, makes xe-0/0/0.200 a member of that VLAN, enables Q-in-Q tunneling on interface xe-0/0/0, enables xe-0/0/0 to accept untagged packets, and binds a logical interface of xe-0/0/0 to the VLAN ID of VLAN v200.

```
set vlans v200 interface xe-0/0/0.200
set interfaces xe-0/0/0 flexible-vlan-tagging
set interfaces xe-0/0/0 native-vlan-id 150
set interfaces xe-0/0/0 encapsulation extended-vlan-bridge
set interfaces xe-0/0/0 unit 200 vlan-id 200
```

Now configure a specific interface mapping with optional VLAN ID translation on the C-VLAN interface:

1. Assign a logical interface of the C-VLAN interface to be a member of the S-VLAN.

   [edit vlans *vlan-name*]
   `user@switch#` **set interface *interface-name.unit-number***

2. Enable the interface to transmit packets with 802.1Q VLAN tags:

   [edit interfaces *interface-name*]
   `user@switch#` **set** flexible-vlan-tagging

3. Enable the C-VLAN interface to send and receive untagged packets:

   [edit interfaces *interface-name*]
   `user@switch#` **set native-vlan-id *vlan-id***

4. Enable extended VLAN bridge encapsulation on the interface:

   [edit interfaces *interface-name*]
   `user@switch#` **set** encapsulation **extended-vlan-bridge**

5. Configure a logical interface (unit) to receive and forward any tagged packet whose VLAN ID tag matches the VLAN IDs you specify:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set vlan-id *number***

6. Configure the system to remove the existing C-VLAN tag and replace it with the S-VLAN tag when packets ingress on the C-VLAN interface and are forwarded to the S-VLAN:

   [edit interfaces *interface-name* unit *logical-unit-number*]
   `user@switch#` **set** input-vlan-map swap

7. Configure the system to remove the existing S-VLAN tag and replace it with the C-VLAN tag when packets are forwarded from the S-VLAN interface to the C-VLAN interface:

   [edit interfaces *interface-name* unit *logical-unit-number*]

```
user@switch# set output-vlan-map swap
```

8. To configure an S-VLAN and associate it with the appropriate C-VLAN interface:

```
[edit vlans vlan-name]
user@switch# set interface interface-name
```

For example, the following configuration on C-VLAN interface xe-0/0/1.200 enables Q-in-Q tunneling, enables xe-0/0/1 to accept untagged packets, and maps incoming packets from C-VLAN 150 to logical interface 200, which is a member of S-VLAN 200. Also, when packets egress from C-VLAN interface xe-0/0/1 and travel to the S-VLAN interface, the C-VLAN tag of 150 is removed and replaced with the S-VLAN tag of 200. When packets travel from the S-VLAN interface to the C-VLAN interface, the S-VLAN tag of 200 is removed and replaced with the C-VLAN tag of 150.

```
set vlans v200 interface xe-0/0/1.200
set interfaces xe-0/0/1 flexible-vlan-tagging
set interfaces xe-0/0/1 native-vlan-id 150
set interfaces xe-0/0/1 encapsulation extended-vlan-bridge
set interfaces xe-0/0/1 unit 200 vlan-id 200
set interfaces xe-0/0/1 unit 200 output-vlan-map swap
set interfaces xe-0/0/1 unit 200 input-vlan-map swap
```

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using All-in-One Bundling on page 565
- Configuring Q-in-Q Tunneling Using Many-to-Many Bundling on page 567

Example: Setting Up Q-in-Q Tunneling on QFX Series Switches

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 Switches" on page 93.

Overview and Topology

In this service provider network, there are multiple customer VLANs mapped to one service VLAN.

Table 88 on page 599 lists the settings for the sample topology.

Table 88: 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
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 554
- Configuring Q-in-Q Tunneling on QFX Series Switches on page 581
- *Troubleshooting Q-in-Q and VLAN Translation Configuration*

Example: Setting Up Q-in-Q Tunneling on EX Series Switches

Service providers can use Q-in-Q tunneling to transparently pass Layer 2 VLAN traffic from a customer site, through the service provider network, to another customer site without removing or changing the customer VLAN tags or class-of-service (CoS) settings. You can configure Q-in-Q tunneling on EX Series switches.

This example describes how to set up Q-in-Q:

Requirements

This example requires one EX Series switch with Junos OS Release 9.3 or later for EX Series switches.

Before you begin setting up Q-in-Q tunneling, make sure you have created and configured the necessary customer VLANs. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98 or *Configuring VLANs for EX Series Switches (J-Web Procedure)*.

Overview and Topology

In this service provider network, there are multiple customer VLANs mapped to one service VLAN.

Table 88 on page 599 lists the settings for the example topology.

Table 89: Components of the Topology for Setting Up Q-in-Q Tunneling

Interface	Description
ge-0/0/11.0	Tagged S-VLAN trunk port
ge-0/0/12.0	Untagged customer-facing access port
ge-0/0/13.0	Untagged customer-facing access port
ge-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 qinqvlan vlan-id 4001
set vlans qinqvlan dot1q-tunneling customer-vlans 1-100
set vlans qinqvlan dot1q-tunneling customer-vlans 201-300
set interfaces ge-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/11 unit 0 family ethernet-switching vlan members 4001
set interfaces ge-0/0/12 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/12 unit 0 family ethernet-switching vlan members 4001
set interfaces ge-0/0/13 unit 0 family ethernet-switching port-mode access
set interfaces ge-0/0/13 unit 0 family ethernet-switching vlan members 4001
set interfaces ge-0/0/14 unit 0 family ethernet-switching port-mode trunk
set interfaces ge-0/0/14 unit 0 family ethernet-switching vlan members 4001
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 qinqvlan vlan-id 4001
```

2. Enable Q-in-Q tuennling and specify the customer VLAN ranges:

```
[edit vlans]
user@switch# set qinqvlan dot1q-tunneling customer-vlans 1-100
user@switch# set qinqvlan dot1q-tunneling customer-vlans 201-300
```

3. Set the port mode and VLAN information for the interfaces:

```
[edit interfaces]
user@switch# set ge-0/0/11 unit 0 family ethernet-switching port-mode trunk
user@switch# set ge-0/0/11 unit 0 family ethernet-switching vlan members 4001
user@switch# set ge-0/0/12 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/12 unit 0 family ethernet-switching vlan members 4001
user@switch# set ge-0/0/13 unit 0 family ethernet-switching port-mode access
user@switch# set ge-0/0/13 unit 0 family ethernet-switching vlan members 4001
user@switch# set ge-0/0/14 unit 0 family ethernet-switching port-mode trunk
user@switch# set ge-0/0/14 unit 0 family ethernet-switching vlan members 4001
```

4. Set the Q-in-Q Ethertype value:

```
[edit]
user@switch# set ethernet-switching-options dot1q-tunneling ether-type 0x9100
```

Results

Check the results of the configuration:

```
user@switch> show configuration vlans qinqvlan
vlan-id 4001 {
    dot1q-tunneling {
    customer-vlans [ 1-100 201-300 ];
}
user@switch> show configuration interfaces
ge-0/0/11 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan members 4001;
        }
    }
}
ge-0/0/12 {
    unit 0 {
        family ethernet-switching {
            port-mode access;
            vlan members 4001;
        }
    }
```

```
}
ge-0/0/13 {
  unit 0 {
    family ethernet-switching {
      port-mode access;
      vlan members 4001;
    }
  }
}
ge-0/0/14 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan members 4001;
    }
  }
}
user@switch> show ethernet-switching-options
dot1q-tunneling {
  ether-type 0x9100;
}
```

Verification

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

- Verifying That Q-in-Q Tunneling Was Enabled on page 604

Verifying That Q-in-Q Tunneling Was Enabled

Purpose Verify that Q-in-Q tunneling was properly enabled on the switch.

Action Use the **show vlans** command:

```
user@switch>  show vlans qinqvlan extensive
VLAN: qinqvlan, Created at: Thu Sep 18 07:17:53 2008
802.1Q Tag: 4001, 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  4 (Active = 0)
      ge-0/0/11.0, tagged, trunk
      ge-0/0/14.0, tagged, trunk
      ge-0/0/12.0, untagged, access
      ge-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

- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582

Setting Up a Dual VLAN Tag Translation Configuration on QFX Switches

Starting with Junos OS Release 14.1X53-D40, you can use the dual VLAN tag translation (also known as dual VLAN tag rewrite) feature to deploy switches in service-provider domains, allowing dual-tagged, single-tagged, and untagged VLAN packets to come into or exit from the switch.

The following example configuration shows use of the swap-swap, pop-swap, and swap-push dual tag operations.

```
[edit]
set interfaces ge-0/0/1 unit 503 description UNI-3
set interfaces ge-0/0/1 unit 503 encapsulation vlan-bridge
set interfaces ge-0/0/1 unit 503 vlan-tags outer 503
set interfaces ge-0/0/1 unit 503 vlan-tags inner 504
set interfaces ge-0/0/1 unit 503 input-vlan-map swap-swap
set interfaces ge-0/0/1 unit 503 input-vlan-map vlan-id 500
set interfaces ge-0/0/1 unit 503 input-vlan-map inner-vlan-id 514
set interfaces ge-0/0/1 unit 503 output-vlan-map swap-swap
set interfaces ge-0/0/0 description NNI
set interfaces ge-0/0/0 flexible-vlan-tagging
set interfaces ge-0/0/0 encapsulation flexible-ethernet-services
set interfaces ge-0/0/0 unit 500 description "SVLAN500 port"
set interfaces ge-0/0/0 unit 500 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 500 vlan-id 500
set interfaces ge-0/0/0 unit 600 description "SVLAN600 port"
set interfaces ge-0/0/0 unit 600 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 600 vlan-id 600
set interfaces ge-0/0/0 unit 700 description "SVLAN700 port"
set interfaces ge-0/0/0 unit 700 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 700 vlan-id 700
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members v1000
set interfaces ge-0/0/0 unit 1100 description UNI-SVLAN1100
set interfaces ge-0/0/0 unit 1100 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 1100 vlan-tags outer 1101
set interfaces ge-0/0/0 unit 1100 vlan-tags inner 1102
set interfaces ge-0/0/0 unit 1100 input-vlan-map swap-swap
set interfaces ge-0/0/0 unit 1100 input-vlan-map vlan-id 1100
set interfaces ge-0/0/0 unit 1100 input-vlan-map inner-vlan-id 2101
set interfaces ge-0/0/0 unit 1100 output-vlan-map swap-swap
set interfaces ge-0/0/0 unit 1200 description UNI-SVLAN1200
set interfaces ge-0/0/0 unit 1200 encapsulation vlan-bridge
set interfaces ge-0/0/0 unit 1200 vlan-id 1201
set interfaces ge-0/0/0 unit 1200 input-vlan-map swap-push
set interfaces ge-0/0/0 unit 1200 input-vlan-map inner-vlan-id 2200
set interfaces ge-0/0/0 unit 1200 output-vlan-map pop-swap
set interfaces ge-0/0/2 description UNI
set interfaces ge-0/0/2 flexible-vlan-tagging
set interfaces ge-0/0/2 encapsulation flexible-ethernet-services
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members v1000
set interfaces ge-0/0/2 unit 603 description UNI-3
set interfaces ge-0/0/2 unit 603 encapsulation vlan-bridge
set interfaces ge-0/0/2 unit 603 vlan-tags outer 603
```

```
set interfaces ge-0/0/2 unit 603 vlan-tags inner 604
set interfaces ge-0/0/2 unit 603 input-vlan-map swap-swap
set interfaces ge-0/0/2 unit 603 input-vlan-map vlan-id 600
set interfaces ge-0/0/2 unit 603 input-vlan-map inner-vlan-id 614
set interfaces ge-0/0/2 unit 603 output-vlan-map swap-swap
set interfaces ge-0/0/3 description UNI
set interfaces ge-0/0/3 flexible-vlan-tagging
set interfaces ge-0/0/3 encapsulation flexible-ethernet-services
set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members v1000
set interfaces ge-0/0/3 unit 703 description UNI-3
set interfaces ge-0/0/3 unit 703 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 703 vlan-tags outer 703
set interfaces ge-0/0/3 unit 703 vlan-tags inner 704
set interfaces ge-0/0/3 unit 703 input-vlan-map swap-swap
set interfaces ge-0/0/3 unit 703 input-vlan-map vlan-id 700
set interfaces ge-0/0/3 unit 703 input-vlan-map inner-vlan-id 714
set interfaces ge-0/0/3 unit 703 output-vlan-map swap-swap
set interfaces ge-0/0/3 unit 701 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 701 vlan-id 701
set interfaces ge-0/0/3 unit 701 input-vlan-map swap-push
set interfaces ge-0/0/3 unit 701 input-vlan-map inner-vlan-id 780
set interfaces ge-0/0/3 unit 701 output-vlan-map pop-swap
set interfaces ge-0/0/3 unit 1100 description SVLAN1100-NNI
set interfaces ge-0/0/3 unit 1100 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 1100 vlan-id 1100
set interfaces ge-0/0/3 unit 1200 description SVLAN1200-NNI
set interfaces ge-0/0/3 unit 1200 encapsulation vlan-bridge
set interfaces ge-0/0/3 unit 1200 vlan-id 1200
set vlans SVLAN500 interface ge-0/0/0.500
set vlans SVLAN500 interface ge-0/0/1.503
set vlans SVLAN600 interface ge-0/0/0.600
set vlans SVLAN600 interface ge-0/0/2.603
set vlans SVLAN600 interface ge-0/0/3.701
set vlans SVLAN700 interface ge-0/0/0.700
set vlans SVLAN700 interface ge-0/0/3.703
set vlans v1000 vlan-id 1000
set vlans SVLAN1100 interface ge-0/0/0.1100
set vlans SVLAN1100 interface ge-0/0/3.1100
set vlans SVLAN1200 interface ge-0/0/3.1200
set vlans SVLAN1200 interface ge-0/0/0.1200
```

Release History Table

Release	Description
14.1X53-D40	Starting with Junos OS Release 14.1X53-D40, you can use the dual VLAN tag translation (also known as dual VLAN tag rewrite) feature to deploy switches in service-provider domains, allowing dual-tagged, single-tagged, and untagged VLAN packets to come into or exit from the switch.

Related Documentation

- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling Using All-in-One Bundling on page 565
- Configuring Q-in-Q Tunneling Using Many-to-Many Bundling on page 567

Verifying That Q-in-Q Tunneling Is Working on Switches

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 on EX Series Switches (CLI Procedure) on page 582
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601

CHAPTER 22

Configuring Redundant Trunk Groups

Understanding Redundant Trunk Links (Legacy RTG Configuration)

In a typical enterprise network composed of distribution and access layers, a redundant trunk link provides a simple solution for network recovery when a trunk port on a switch goes down. In that case, traffic is routed to another trunk port, keeping network convergence time to a minimum.

NOTE: For information on redundant trunk link configurations that include Q-in-Q support and use LAGs with link protection, see *Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection*.

To configure a redundant trunk link, create a redundant trunk group. The redundant trunk group is configured on the access switch and contains two links: a primary or active link, and a secondary link. If the active link fails, the secondary link automatically starts forwarding data traffic without waiting for normal spanning-tree protocol convergence.

Data traffic is forwarded only on the active link. Data traffic on the secondary link is dropped and shown as dropped packets when you issue the operational mode command **show interfaces *interface-name* extensive**.

While data traffic is blocked on the secondary link, Layer 2 control traffic is still permitted. For example, an LLDP session can be run between two switches on the secondary link.

Rapid Spanning Tree Protocol (RSTP) is enabled by default on the switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time. You must disable RSTP on an interface if a redundant trunk group is configured on that interface. For example, in Figure 35 on page 611, in addition to disabling RSTP on the Switch 3 interfaces, you must also disable RSTP on the Switch 1 and Switch 2 interfaces connected to Switch 3. Spanning-tree protocols can, however, continue operating on other interfaces on those switches—for example on the link between Switch 1 and Switch 2.

Figure 35 on page 611 shows three switches in a basic topology for redundant trunk links. Switch 1 and Switch 2 make up the distribution layer, and Switch 3 makes up the access layer. Switch 3 is connected to the distribution layer through trunk ports ge-0/0/9.0 (Link 1) and ge-0/0/10.0 (Link 2). Link 1 and Link 2 are in a redundant trunk group called group1. Link 1 is designated as the primary link. Traffic flows between Switch 3 in the access layer and Switch 1 in the distribution layer through Link 1. While Link 1 is active, Link 2 blocks traffic.

Figure 35: Redundant Trunk Group, Link 1 Active

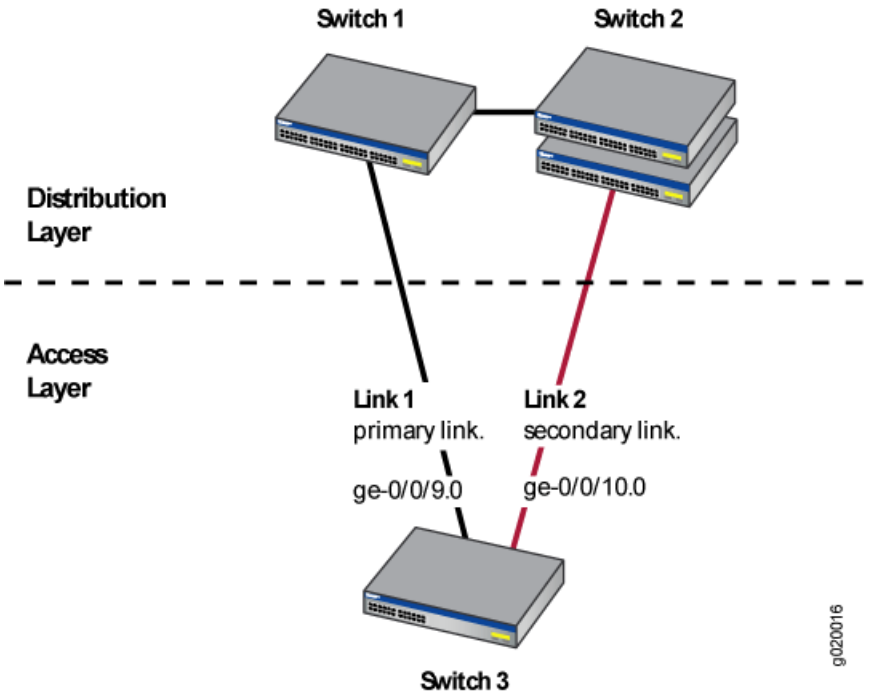
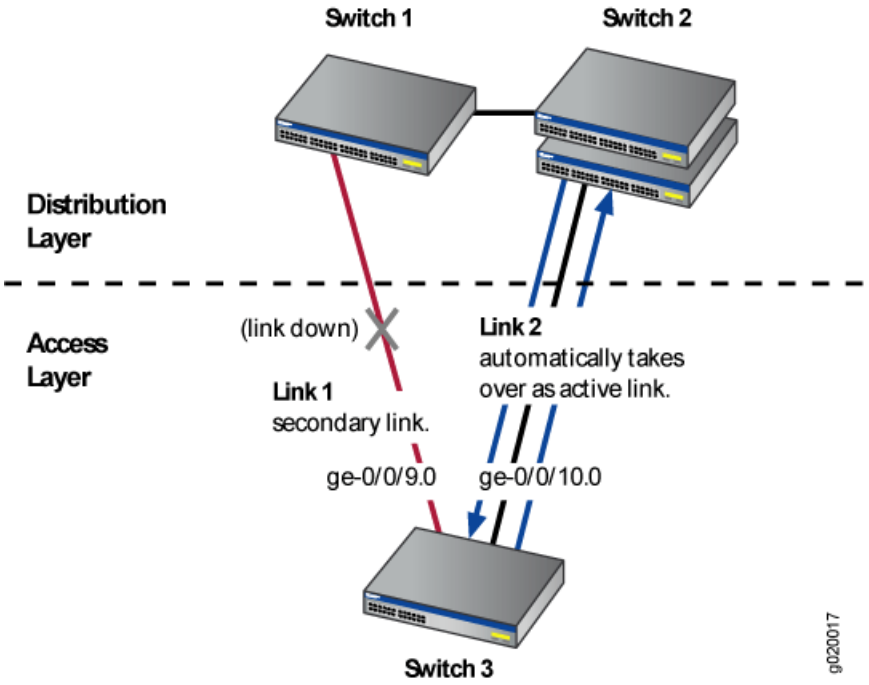


Figure 36 on page 611 illustrates how the redundant trunk link topology works when the primary link goes down.

Figure 36: Redundant Trunk Group, Link 2 Active



When Link 1 between Switch 1 and Switch 3 goes down, Link 2 takes over as the active link. Traffic between the access layer and the distribution layer is then automatically switched to Link 2 between Switch 3 and Switch 2.

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support on page 613

Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches (CLI Procedure)

You can manage network convergence by configuring both a primary link and a secondary link on an EX Series switch; this is called a redundant trunk group (RTG). If the primary link in a redundant trunk group fails, it passes its known MAC address locations to the secondary link, which automatically takes over. You can configure a maximum of 16 redundant trunk groups on most standalone switches or on Virtual Chassis. The EX8200 switch and EX8200 Virtual Chassis, however, support up to 254 redundant trunk groups.

Generally, you configure a redundant trunk group by configuring one primary link (and its interface) and one unspecified link (and its interface) to serve as the secondary link. A second type of redundant trunk group, not shown in the procedure in this topic, consists of two unspecified links (and their interfaces); in this case, neither of the links is primary. In this second case, the software selects an active link by comparing the port numbers of the two links and activating the link with the higher port number. The procedure given here describes configuring a primary/unspecified configuration for a redundant trunk group because that configuration gives you more control and is more commonly used.

Rapid Spanning Tree Protocol (RSTP) is enabled by default on EX Series switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time.

A primary link takes over whenever it is able. You can, however, alter the number of seconds that the primary link waits before reestablishing control by configuring the primary link's preempt cutover timer.

Before you configure the redundant trunk group on the switch, be sure you have:

- Disabled RSTP on all switches that will be linked to your redundant trunk group.
- Configured at least two interfaces with their port mode set to **trunk**; be sure that these two interfaces are not part of any existing RTG. See *Configuring Gigabit Ethernet Interfaces (CLI Procedure)* .

To configure a redundant trunk group on a switch:

1. Turn off RSTP:

```
[edit]
user@switch# set protocols rstp disable
```

2. Name the redundant trunk group while configuring one primary and one unspecified trunk interface:

```
[edit ethernet-switching-options]
user@switch# set redundant-trunk-group group name interface interface-name primary
user@switch# set redundant-trunk-group group name interface interface-name
```

3. (Optional) Change the length of time (from the default of 1 second) that a re-enabled primary link waits to take over from an active secondary link:

```
[edit ethernet-switching-options]
set redundant-trunk-group group name preempt-cutover-timer seconds
```

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support

NOTE: This example uses Junos OS for EX Series switches or QFX Series with support for the Enhanced Layer 2 Software (ELS) configuration style.. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

You can manage network convergence by configuring both a primary link and a secondary link on a switch; this is called a redundant trunk group (RTG). If the primary link in a redundant trunk group fails, it passes its known MAC address locations to the secondary link, which automatically takes over after one minute.

This example describes how to create a redundant trunk group with a primary and a secondary link:

Requirements

This example uses the following hardware and software components:

- Two EX Series or QFX Series distribution switches
- One EX Series or QFX Series access switch
- The appropriate software release for your platform:

- For EX Series switches: Junos OS Release 13.2X50-D10 or later
- For the QFX Series: Junos OS Release 13.2X50-D15 or later

Before you configure the redundant trunk links network on the access and distribution switches, be sure you have:

- Configured interfaces ge-0/0/9 and ge-0/0/10 on the access switch, Switch 3, as trunk interfaces.
- Configured one trunk interface on each distribution switch, Switch 1 and Switch 2.
- Connected the three switches as shown in the topology for this example (see Figure 37 on page 616).

Overview and Topology

In a typical enterprise network composed of distribution and access layers, a redundant trunk link provides a simple solution for trunk interface network recovery. When a trunk interface fails, data traffic is routed to another trunk interface after one minute, thereby keeping network convergence time to a minimum.

This example shows the configuration of a redundant trunk group that includes one primary link (and its interface) and one unspecified link (and its interface) that serves as the secondary link.

A second type of redundant trunk group, not illustrated in the example, consists of two unspecified links (and their interfaces); in this case, neither of the links is primary. The software selects an active link by comparing the port numbers of the two links and activating the link with the higher port number. For example, if the two link interfaces use interfaces ge-0/1/0 and ge-0/1/1, the software activates ge-0/1/1. (In the interface names, the final number is the port number.)

The two links in a redundant trunk group generally operate the same way, whether they are configured as primary/unspecified or unspecified/unspecified. Data traffic initially passes through the active link but is blocked on the inactive link. While data traffic is blocked on the secondary link, note that Layer 2 control traffic is still permitted if the link is active. For example, an LLDP session can be run between two switches on the secondary link. If the active link either goes down or is disabled administratively, it broadcasts a list of its known MAC addresses for data traffic; the other link immediately picks up and adds the MAC addresses to its address table, becomes active, and begins forwarding traffic.

The one difference in operation between the two types of redundant trunk groups occurs when a primary link is active, goes down, is replaced by the secondary link, and then reactivates. When a primary link is re-enabled while the secondary link is active, the primary link waits 1 second (you can change the time interval by using the preempt cutover timer to accommodate your network) and then takes over as the active link. In other words, the primary link has priority and is always activated if it is available. This differs from the behavior of two unspecified links, both of which act as equals. Because the unspecified links are equal, the active link remains active until it either goes down or is disabled administratively; this is the only time that the other unspecified link learns the MAC addresses and immediately becomes active.

The example given here illustrates a primary/unspecified configuration for a redundant trunk group because that configuration gives you more control and is more commonly used.

NOTE: Rapid Spanning Tree Protocol (RSTP) is enabled by default on the switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time. You will need to disable RSTP on the two distribution switches in the example, Switch 1 and Switch 2. Spanning-tree protocols can, however, continue operating in other parts of the network—for example, between the distribution switches and also in links between distribution switches and the enterprise core.

Figure 37 on page 616 displays an example topology containing three switches. Switch 1 and Switch 2 make up the distribution layer, and Switch 3 makes up the access layer. Switch 3 is connected to the distribution layer through trunk interfaces ge-0/0/9.0 (Link 1) and ge-0/0/10.0 (Link 2).

Table 90 on page 616 lists the components used in this redundant trunk group.

Because RSTP and RTGs cannot operate simultaneously on a switch, you disable RSTP on Switch 1 and Switch 2 in the first configuration task, and you disable RSTP on Switch 3 in the second task.

The second configuration task creates a redundant trunk group called example 1 on Switch 3. The trunk interfaces ge-0/0/9.0 and ge-0/0/10.0 are the two links configured in the second configuration task. You configure the trunk interface ge-0/0/9.0 as the primary link. You configure the trunk interface ge-0/0/10.0 as an unspecified link, which becomes the secondary link by default.

Figure 37: Topology for Configuring the Redundant Trunk Links

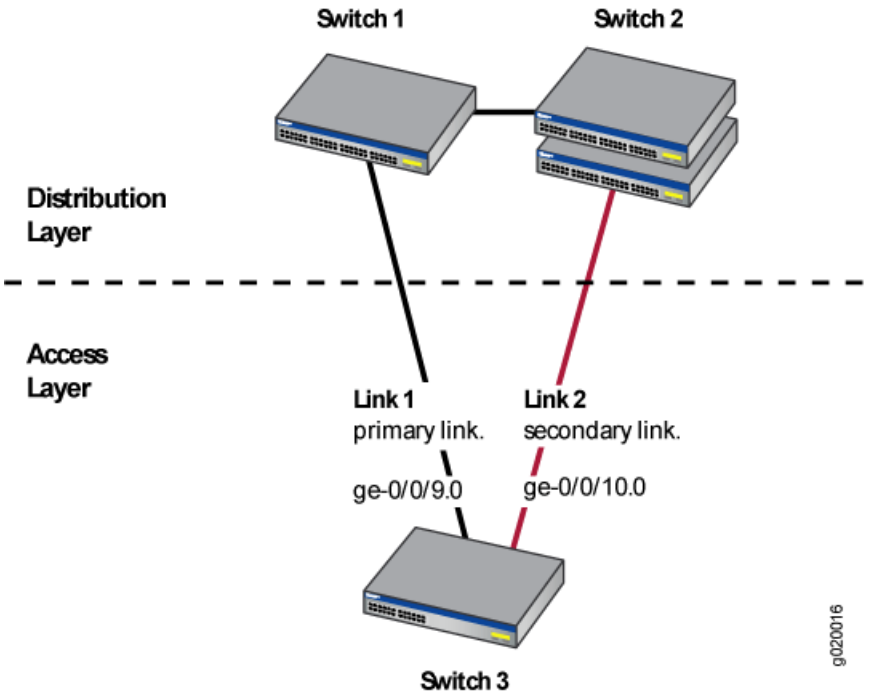


Table 90: Components of the Redundant Trunk Link Topology

Property	Settings
Switch hardware	• Switch 1–1 EX Series or QFX Series distribution switch • Switch 2–1 EX Series or QFX Series distribution switch • Switch 3–1 EX Series or QFX Series access switch
Trunk interfaces	On Switch 3 (access switch): ge-0/0/9.0 and ge-0/0/10.0
Redundant trunk group	rtg0

Disabling RSTP on Switches 1 and 2

To disable RSTP on Switch 1 and Switch 2, perform this task on each switch:

CLI Quick Configuration

To quickly disable RSTP on Switch 1 and Switch 2, copy the following command and paste it into each switch terminal window:

```
[edit]
set protocols rstp disable
```

Step-by-Step Procedure

To disable RSTP on Switch 1 and Switch 2:

1. Disable RSTP on Switch 1 and Switch 2:

```
[edit]
user@switch# set protocols rstp disable
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
protocols {
  rstp {
    disable;
  }
}
```

Configuring Redundant Trunk Links on Switch 3

To configure redundant trunk links on Switch 3, perform this task:

CLI Quick Configuration

To quickly configure the redundant trunk group rtg0 on Switch 3, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols rstp disable
set switch-options redundant-trunk-group group rtg0 interface ge-0/0/9.0 primary
set switch-options redundant-trunk-group group rtg0 interface ge-0/0/10.0
set redundant-trunk-group group rtg0 preempt-cutover-timer 60
```

Step-by-Step Procedure

Configure the redundant trunk group rtg0 on Switch 3.

1. Turn off RSTP:

```
[edit]
user@switch# set protocols rstp disable
```

2. Name the redundant trunk group rtg0 while configuring trunk interface ge-0/0/9.0 as the primary link and ge-0/0/10 as an unspecified link to serve as the secondary link:

```
[edit switch-options]
user@switch#  set redundant-trunk-group group rtg0 interface ge-0/0/9.0 primary
user@switch#  set redundant-trunk-group group rtg0 interface ge-0/0/10.0
```

3. (Optional) Change the time interval (from the default of 1 second) that a re-enabled primary link waits to take over for an active secondary link:

```
[edit switch-options]
user@switch#  set redundant-trunk-group group rtg0 preempt-cutover-timer 60
```

Results Check the results of the configuration:

```
[edit]
user@switch# show
switch-options
  redundant-trunk-group {
    group rtg0 {
      preempt-cutover-timer 60;
      interface ge-0/0/9.0 {
        primary;
      }
      interface ge-0/0/10.0;
    }
  }
protocols {
  rstp {
    disable;
  }
}
```

Verification

To confirm that the configuration is set up correctly, perform this task:

- Verifying That a Redundant Trunk Group Was Created on page 618

Verifying That a Redundant Trunk Group Was Created

Purpose Verify that the redundant trunk group rtg0 has been created on Switch 1 and that trunk interfaces are members of the redundant trunk group.

Action List all redundant trunk groups configured on the switch:

```
user@switch> show redundant-trunk-group
Group       Interface   State       Time of last flap          Flap
name                                                           count

rtg0        ge-0/0/9.0  Up/Pri      Never                        0
            ge-0/0/10.0 Up          Never                        0
```

Meaning The **show redundant-trunk-group** command lists all redundant trunk groups configured on the switch as well as the interface names and their current states (up or down for an unspecified link, and up or down and primary for a primary link). For this configuration example, the output shows that the redundant trunk group rtg0 is configured on the switch. The **Up** beside the interfaces indicates that both link cables are physically connected. The **Pri** beside trunk interface ge-0/0/9.0 indicates that it is configured as the primary link.

Related Documentation

- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches

NOTE: This example uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support" on page 613. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

You can manage network convergence by configuring both a primary link and a secondary link on a switch; this is called a redundant trunk group (RTG). If the primary link in a redundant trunk group fails, it passes its known MAC address locations to the secondary link, which automatically takes over after one minute.

This example describes how to create a redundant trunk group with a primary and a secondary link:

Requirements

This example uses the following hardware and software components:

- Two EX Series distribution switches
- One EX Series access switch
- Junos OS Release 10.4 or later for EX Series switches

Before you configure the redundant trunk links network on the access and distribution switches, be sure you have:

- Configured interfaces **ge-0/0/9** and **ge-0/0/10** on the access switch, Switch 3, as trunk interfaces. See *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*.
- Configured one trunk interface on each distribution switch, Switch 1 and Switch 2.
- Connected the three switches as shown in the topology for this example (see Figure 37 on page 616).

Overview and Topology

In a typical enterprise network composed of distribution and access layers, a redundant trunk link provides a simple solution for trunk interface network recovery. When a trunk

interface fails, data traffic is routed to another trunk interface after one minute, thereby keeping network convergence time to a minimum.

This example shows the configuration of a redundant trunk group that includes one primary link (and its interface) and one unspecified link (and its interface) that serves as the secondary link.

A second type of redundant trunk group, not illustrated in the example, consists of two unspecified links (and their interfaces); in this case, neither of the links is primary. In this second case, the software selects an active link by comparing the port numbers of the two links and activating the link with the higher port number. For example, if the two link interfaces use interfaces **ge-0/1/0** and **ge-0/1/1**, the software activates **ge-0/1/1.** (In the interface names, the final number is the port number.)

The two links in a redundant trunk group generally operate the same way, whether they are configured as primary/unspecified or unspecified/unspecified. Data traffic initially passes through the active link but is blocked on the inactive link. While data traffic is blocked on the secondary link, note that Layer 2 control traffic is still permitted if the link is active. For example, an LLDP session can be run between two switches on the secondary link. If the active link either goes down or is disabled administratively, it broadcasts a list of its known MAC addresses for data traffic; the other link immediately picks up and adds the MAC addresses to its address table, becomes active, and begins forwarding traffic.

The one difference in operation between the two types of redundant trunk groups occurs when a primary link is active, goes down, is replaced by the secondary link, and then reactivates. When a primary link is re-enabled while the secondary link is active, the primary link waits 1 second (you can change the length of time using the preempt cutover timer to accommodate your network) and then takes over as the active link. In other words, the primary link has priority and is always activated if it is available. This differs from the behavior of two unspecified links, which act as equals. Because the unspecified links are equal, the active link remains active until it either goes down or is disabled administratively; this is the only time that the other unspecified link learns the MAC addresses and immediately becomes active.

The example given here illustrates a primary/unspecified configuration for a redundant trunk group because that configuration gives you more control and is more commonly used.

NOTE: Rapid Spanning Tree Protocol (RSTP) is enabled by default on EX Series switches to create a loop-free topology, but an interface is not allowed to be in both a redundant trunk group and in a spanning-tree protocol topology at the same time. You will need to disable RSTP on the two distribution switches in the example, Switch 1 and Switch 2. Spanning-tree protocols can, however, continue operating in other parts of the network—for example, between the distribution switches and also in links between distribution switches and the enterprise core.

Figure 37 on page 616 displays an example topology containing three switches. Switch 1 and Switch 2 make up the distribution layer, and Switch 3 makes up the access layer.

Switch 3 is connected to the distribution layer through trunk interfaces **ge-0/0/9.0** (Link 1) and **ge-0/0/10.0** (Link 2).

Table 90 on page 616 lists the components used in this redundant trunk group.

Because RSTP and RTGs cannot operate simultaneously on a switch, you disable RSTP on Switch 1 and Switch 2 in the first configuration task, and you disable RSTP on Switch 3 in the second task.

The second configuration task creates a redundant trunk group called **example 1** on Switch 3. The trunk interfaces **ge-0/0/9.0** and **ge-0/0/10.0** are the two links configured in the second configuration task. You configure the trunk interface **ge-0/0/9.0** as the primary link. You configure the trunk interface **ge-0/0/10.0** as an unspecified link, which becomes the secondary link by default.

Figure 38: Topology for Configuring the Redundant Trunk Links

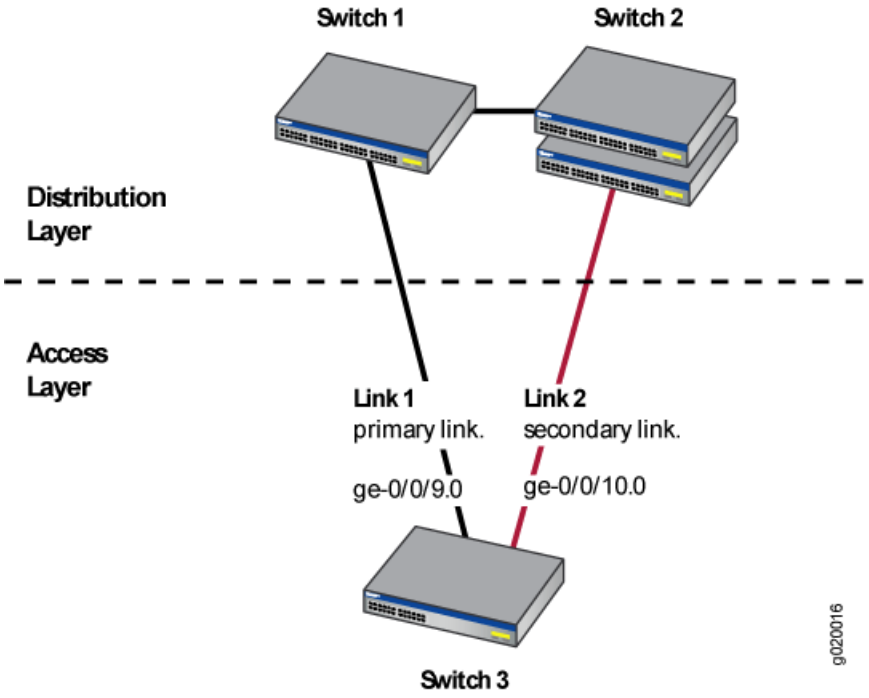


Table 91: Components of the Redundant Trunk Link Topology

Property	Settings
Switch hardware	• Switch 1–1 EX Series distribution switch • Switch 2–1 EX Series distribution switch • Switch 3–1 EX Series access switch
Trunk interfaces	On Switch 3 (access switch): **ge-0/0/9.0** and **ge-0/0/10.0**
Redundant trunk group	**example1**

Disabling RSTP on Switches 1 and 2

To disable RSTP on Switch 1 and Switch 2, perform this task on each switch:

CLI Quick Configuration

To quickly disable RSTP on Switch 1 and Switch 2, copy the following command and paste it into each switch terminal window:

```
[edit]
set protocols rstp disable
```

Step-by-Step Procedure

To disable RSTP on Switch 1 and Switch 2:

1. Disable RSTP on Switch 1 and Switch 2:

```
[edit]
user@switch# set protocols rstp disable
```

Results

Check the results of the configuration:

```
[edit]
user@switch# show
protocols {
    rstp {
        disable;
    }
}
```

Configuring Redundant Trunk Links on Switch 3

To configure redundant trunk links on Switch 3, perform this task:

CLI Quick Configuration

To quickly configure the redundant trunk group **example1** on Switch 3, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols rstp disable
set ethernet-switching-options redundant-trunk-group group example1 interface ge-0/0/9.0
primary
set ethernet-switching-options redundant-trunk-group group example1 interface ge-0/0/10.0
set ethernet-switching-options redundant-trunk-group group example1 preempt-cutover-timer
60
```

Step-by-Step Procedure

Configure the redundant trunk group **example1** on Switch 3.

1. Turn off RSTP:

```
[edit]
user@switch# set protocols rstp disable
```

2. Name the redundant trunk group example1 while configuring trunk interface **ge-0/0/9.0** as the primary link and **ge-0/0/10** as an unspecified link to serve as the secondary link:

```
[edit ethernet-switching-options]
user@switch#  set redundant-trunk-group group example1 interface ge-0/0/9.0 primary
user@switch#  set redundant-trunk-group group example1 interface ge-0/0/10.0
```

3. (Optional) Change the length of time (from the default of 1 second) that a re-enabled primary link waits to take over for an active secondary link:

```
[edit ethernet-switching-options]
user@switch#  set redundant-trunk-group group example1 preempt-cutover-timer 60
```

Results Check the results of the configuration:

```
[edit]
user@switch# show
ethernet-switching-options
  redundant-trunk-group {
    group example1 {
      preempt-cutover-timer 60;
      interface ge-0/0/9.0 {
        primary;
      }
      interface ge-0/0/10.0;
    }
  }
protocols {
  rstp {
    disable;
  }
}
```

Verification

To confirm that the configuration is set up correctly, perform this task:

- Verifying That a Redundant Trunk Group Was Created on page 623

Verifying That a Redundant Trunk Group Was Created

Purpose Verify that the redundant trunk group **example1** has been created on Switch 1 and that trunk interfaces are members of the redundant trunk group.

Action List all redundant trunk groups configured on the switch:

```
user@switch> show redundant-trunk-group
Group      Interface   State       Time of last flap          Flap
name                                                          count

example1   ge-0/0/9.0  Up/Pri      Never                        0
```

```
                    ge-0/0/10.0 Up          Never                       0
```

Meaning The **show redundant-trunk-group** command lists all redundant trunk groups configured on the switch, both links' interface addresses, and the links' current states (up or down for an unspecified link, and up or down and primary for a primary link). For this configuration example, the output shows that the redundant trunk group **example1** is configured on the switch. The **(Up)** beside the interfaces indicates that both link cables are physically connected. The **(Pri)** beside trunk interface **ge-0/0/9.0** indicates that it is configured as the primary link.

Related Documentation

- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

CHAPTER 23

Configuring 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 an integrated routing and bridging (IRB) interface named irb or a routed VLAN interface (RVI) named vlan. (On EX Series switches that use Juniper Networks Junos operating system (Junos OS) with support for the Enhanced Layer 2 Software (ELS) configuration style, the feature is known as an IRB interface. On EX Series switches that use Junos OS that does not support ELS, the feature is known as an 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 or IRB interfaces.
- If you set proxy ARP to unrestricted, disable gratuitous ARP requests on each interface enabled for proxy ARP.

Related Documentation

- Configuring Proxy ARP on Switches on page 627
- **proxy-arp on page 1087**
- Example: Configuring Proxy ARP on an EX Series Switch on page 629
- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 628

Configuring Proxy ARP on Switches

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 625
- Verifying That Proxy ARP Is Working Correctly on page 634
- Understanding Integrated Routing and Bridging on page 445

Configuring Proxy ARP on Switches (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches that does not support the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see "Configuring Proxy ARP on Devices with ELS Support (CLI Procedure)" on page 628. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

You can configure proxy Address Resolution Protocol (ARP) on your EX Series switch 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 ge-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

- Example: Configuring Proxy ARP on an EX Series Switch on page 629
- Verifying That Proxy ARP Is Working Correctly on page 634
- Configuring Routed VLAN Interfaces on Switches (CLI Procedure) on page 365

Configuring Proxy ARP on Devices with ELS Support (CLI Procedure)

NOTE: This task uses Junos OS for EX Series switches and QFX3500 and QFX3600 switches with support for the Enhanced Layer 2 Software (ELS) configuration style.If your switch runs software that does not support ELS, see "Configuring Proxy ARP on Switches (CLI Procedure)" on page 627 or "Configuring Proxy ARP on Switches" on page 627. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

You can configure proxy Address Resolution Protocol (ARP) on your switch 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 interface-name unit logical-unit-number proxy-arp (restricted |
unrestricted)
```

BEST PRACTICE: We recommend that you configure proxy ARP in restricted mode. In restricted mode, the switch does not act as a proxy if the source and target IP addresses are on the same subnet. If you decide to use unrestricted mode, disable gratuitous ARP requests on the interface to avoid a situation wherein the switch's response to a gratuitous ARP request appears to the host to be an indication of an IP conflict.

To configure proxy ARP on an integrated routing and bridging (IRB) interface:

```
[edit interfaces]
user@switch# set irb.logical-unit-number proxy-arp restricted
```

Related Documentation

- Example: Configuring Proxy ARP on an EX Series Switch on page 629
- Verifying That Proxy ARP Is Working Correctly on page 634
- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) on page 456

Example: Configuring Proxy ARP on an EX Series Switch

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

This example shows how to configure proxy ARP on an access switch:

Requirements

This example uses the following hardware and software components:

- One EX Series switch
- Junos OS Release 10.0 or later for EX Series switches

Overview and Topology

This example shows the configuration of proxy ARP on an interface of an EX Series switch using restricted mode. In restricted mode, the switch does not act as a proxy for hosts on the same subnet.

The topology for this example consists of one EX Series switch. When a host wants to communicate with a host that is not already in its ARP table, it broadcasts an ARP request for the MAC address of the destination host:

- When proxy ARP is not enabled, a host that shares the same IP address replies directly to the ARP request, providing its MAC address, and future transmissions are sent directly to the destination host MAC address.
- When proxy ARP is enabled, the switch responds to ARP requests, providing the switch's MAC address—even when the destination IP address is the same as the source IP address. Thus, communications must be sent through the switch and then routed through the switch to the appropriate destination.

Configuration

To configure proxy ARP, perform the following tasks:

CLI Quick Configuration

To quickly configure proxy ARP on an interface, copy the following command and paste it into the switch terminal window:

```
[edit]
set interfaces ge-0/0/3 unit 0 proxy-arp restricted
```

Step-by-Step Procedure

You configure proxy ARP on individual interfaces.

1. To configure proxy ARP on an interface:

```
[edit interfaces]
user@switch# set ge-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 does not act as 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 a situation wherein the switch's response to a gratuitous ARP request appears to the host to be an indication of an IP conflict.

```
[edit interfaces]
user@switch# set ge-0/0/3 no-gratuitous-arp-request
```

Results

Display the results of the configuration:

```
user@switch> show configuration
  interfaces {
    ge-0/0/3 {
      unit 0 {
        proxy-arp restricted;
        family ethernet-switching;
      }
    }
```

Verification

To verify that the switch is sending proxy ARP messages, perform these tasks:

- Verifying That the Switch Is Sending Proxy ARP Messages on page 630

Verifying That the Switch Is Sending Proxy ARP Messages

Purpose

Verify that the switch is sending proxy ARP messages.

Action List the system statistics for ARP messages:

```
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 Devices with ELS Support (CLI Procedure) on page 628
- Understanding Proxy ARP on page 625

Restricted and Unrestricted Proxy ARP Overview

By default, the Junos OS responds to an Address Resolution Protocol (ARP) request only if the destination address of the ARP request is local to the incoming interface.

For Ethernet Interfaces, you can configure the router or switches to proxy-reply to the ARP requests using the restricted or unrestricted proxy ARP configuration.

You might want to configure restricted or unrestricted proxy ARP for routers that act as provider edge (PE) devices in Ethernet Layer 2 LAN switching domains.

NOTE: From Junos OS Release 10.0 onward, Junos OS does not respond to proxy ARP requests with the default route 0.0.0.0. This behavior is in compliance with RFC 1027.

Restricted Proxy ARP

Restricted proxy ARP enables the router or switch to respond to the ARP requests in which the physical networks of the source and target are not the same and the router or switch has an active route to the target address in the ARP request. The router does not reply if the target address is on the same subnet and the same interface as the ARP requestor.

Unrestricted Proxy ARP

Unrestricted proxy ARP enables the router or switch to respond to any ARP request, on condition that the router has an active route to the destination address of the ARP request. The route is not limited to the incoming interface of the request, nor is it required to be a direct route.

WARNING: If you configure unrestricted proxy ARP, the proxy router replies to ARP requests for the target IP address on the same interface as the incoming ARP request. This behavior is appropriate for cable modem termination system (CMTS) environments, but might cause Layer 2 reachability problems if you enable unrestricted proxy ARP in other environments.

When an IP client broadcasts the ARP request across the Ethernet wire, the end node with the correct IP address responds to the ARP request and provides the correct MAC address. If the unrestricted proxy ARP feature is enabled, the router response is redundant and might fool the IP client into determining that the destination MAC address within its own subnet is the same as the address of the router.

NOTE: While the destination address can be remote, the source address of the ARP request must be on the same subnet as the interface upon which the ARP request is received. For security reasons, this rule applies to both unrestricted and restricted proxy ARP.

Topology Considerations for Unrestricted Proxy ARP

In most situations, you should not configure the router or switch to perform unrestricted proxy ARP. Do so only for special situations, such as when cable modems are used. Figure 39 on page 633 and Figure 40 on page 633 show examples of situations in which you might want to configure unrestricted proxy ARP.

In Figure 39 on page 633, the edge device is not running any IP protocols. In this case, you configure the core router to perform unrestricted proxy ARP. The edge device is the client of the proxy.

In Figure 40 on page 633, the Broadband Remote Access Server (B-RAS) routers are not running any IP protocols. In this case, you configure unrestricted proxy ARP on the B-RAS interfaces. This allows the core device to behave as though it is directly connected to the end users.

Figure 39: Edge Device Case for Unrestricted Proxy ARP

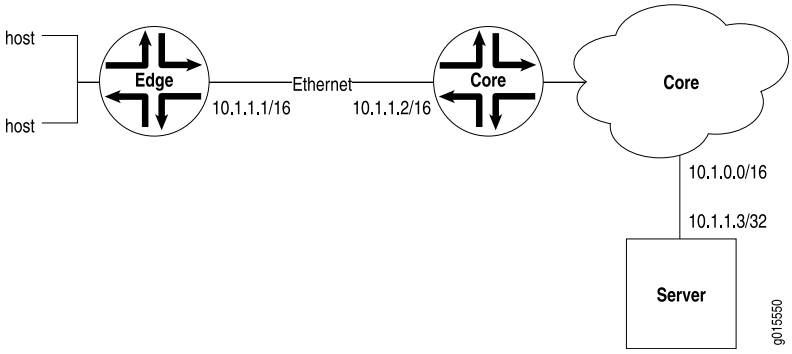
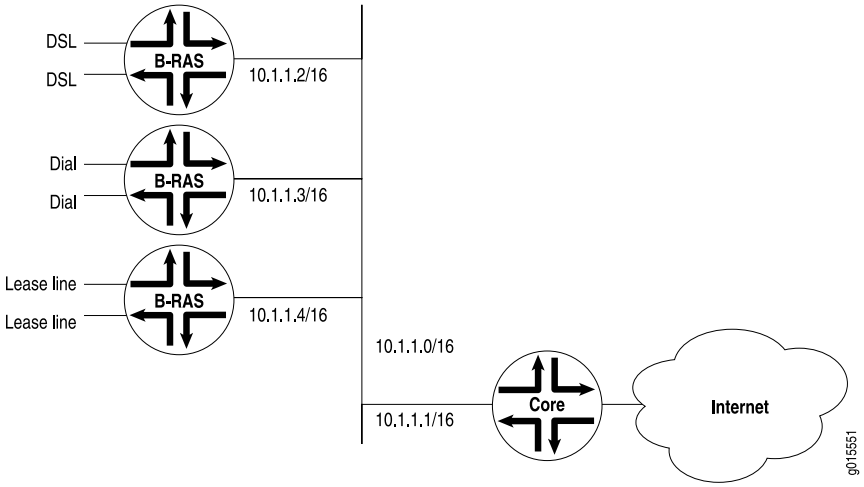


Figure 40: Core Device Case for Unrestricted Proxy ARP



Related Documentation

- Configuring Restricted and Unrestricted Proxy ARP on page 634
- *Ethernet Interfaces Feature Guide for Routing Devices*

Configuring Restricted and Unrestricted Proxy ARP

To configure restricted or unrestricted proxy ARP, include the **proxy-arp** statement:

```
proxy-arp (restricted |unrestricted);
```

You can include this statement at the following hierarchy levels:

- **[edit interfaces** *interface-name* **unit** *logical-unit-number* **]**
- **[edit logical-systems** *logical-system-name* **interfaces** *interface-name* **unit** *logical-unit-number***]**

To return to the default—that is, to disable restricted or unrestricted proxy ARP—delete the **proxy-arp** statement from the configuration:

```
[edit]
user@host# delete interfaces interface-name unit logical-unit-number proxy-arp
```

You can track the number of restricted or unrestricted proxy ARP requests processed by the router or switch by issuing the **show system statistics arp** operational mode command.

NOTE: When proxy ARP is enabled as default or unrestricted, the router or switch responds to any ARP request as long as the device has an active route to the target address of the ARP request. This gratuitous ARP behavior can result in an error when the receiving interface and target response interface are the same and the end device (for example, a client) performs a duplicate address check. To prevent this error, configure the router or switch interface with the **no-gratuitous-arp-reply** statement. See *Configuring Gratuitous ARP* for information about how to disable responses to gratuitous ARP requests.

Related Documentation

- proxy-arp on page 1087
- Restricted and Unrestricted Proxy ARP Overview on page 631
- *Configuring Gratuitous ARP*
- *Ethernet Interfaces Feature Guide for Routing Devices*

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 Switches on page 627
- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 628

CHAPTER 24

Configuring Layer 2 Interfaces on Security Devices

Understanding Layer 2 Interfaces on Security Devices

Layer 2 logical interfaces are created by defining one or more logical units on a physical interface with the family address type **ethernet-switching**. If a physical interface has a **ethernet-switching** family logical interface, it cannot have any other family type in its logical interfaces. A logical interface can be configured in one of the following modes:

- Access mode—Interface accepts untagged packets, assigns the specified VLAN identifier to the packet, and forwards the packet within the VLAN that is configured with the matching VLAN identifier.
- Trunk mode—Interface accepts any packet tagged with a VLAN identifier that matches a specified list of VLAN identifiers. Trunk mode interfaces are generally used to interconnect switches. To configure a VLAN identifier for untagged packets received on the physical interface, use the **native-vlan-id** option. If the **native-vlan-id** option is not configured, untagged packets are dropped.

NOTE: Multiple trunk mode logical interfaces can be defined, as long as the VLAN identifiers of a trunk interface do not overlap with those of another trunk interface. The native-vlan-id must belong to a VLAN identifier list configured for a trunk interface.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Layer 2 Logical Interfaces on Security Devices on page 638

- Layer 2 Transparent Mode Overview on page 377

Example: Configuring Layer 2 Logical Interfaces on Security Devices

This example shows how to configure a Layer 2 logical interface as a trunk port so that the incoming packets can be selectively redirected to a firewall or other security device.

Requirements

Before you begin, configure the VLANs. See "Example: Configuring VLANs on Security Devices" on page 382.

Overview

In this example, you configure logical interface ge-3/0/0.0 as a trunk port that carries traffic for packets tagged with VLAN identifiers 1 through 10; this interface is implicitly assigned to the previously configured VLANs vlan-a and vlan-b. Then you assign a VLAN ID of 10 to any untagged packets received on physical interface ge-3/0/0.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set unit 0 family ethernet-switching interface-mode trunk vlan members 1–10
set vlan-tagging native-vlan-id 10
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a Layer 2 logical interface as a trunk port:

1. Configure the logical interface.

   ```
   [edit interfaces ge-3/0/0]
   user@host# set unit 0 family ethernet-switching interface-mode trunk vlan members
     1–10
   ```

2. Specify a VLAN ID for untagged packets.

   ```
   [edit interfaces ge-3/0/0]
   ```

```
user@host# set vlan-tagging native-vlan-id 10
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show interfaces ge-3/0/0** and **show interfaces ge-3/0/0.0** commands.

Related Documentation

- Understanding Layer 2 Interfaces on Security Devices on page 637
- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Layer 2 Security Zones on page 652

Understanding Mixed Mode (Transparent and Route Mode) on Security Devices

Mixed mode supports both transparent mode (Layer 2) and route mode (Layer 3); it is the default mode. You can configure both Layer 2 and Layer 3 interfaces simultaneously using separate security zones.

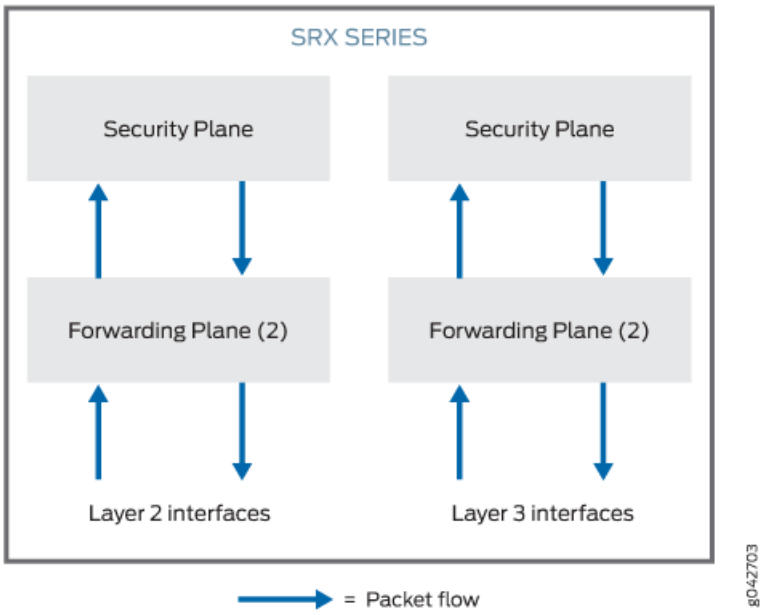
NOTE: For the mixed mode configuration, you must reboot the device after you commit the changes. However, for SRX5000 line devices, reboot is not required.

In mixed mode (Transparent and Route Mode):

- There is no routing among IRB interfaces and between IRB interfaces and Layer 3 interfaces.
- The user logical system is supported for Layer 2 traffic and firewall session function on SRX4100 and SRX4200 devices.
- You can configure Layer 3 interfaces using both the user logical system and the root logical system.

The device in Figure 41 on page 640 looks like two separate devices. One device runs in Layer 2 transparent mode and the other device runs in Layer 3 routing mode. But both devices run independently. Packets cannot be transferred between the Layer 2 and Layer 3 interfaces, because there is no routing among IRB interfaces and between IRB interfaces and Layer 3 interfaces.

Figure 41: Architecture of Mixed Transparent and Route Mode



In mixed mode, the Ethernet physical interface can be either a Layer 2 interface or a Layer 3 interface, but the Ethernet physical interface cannot be both simultaneously. However, Layer 2 and Layer 3 families can exist on separate physical interfaces on the same device.

Table 92 on page 640 lists the Ethernet physical interface types and supported family types.

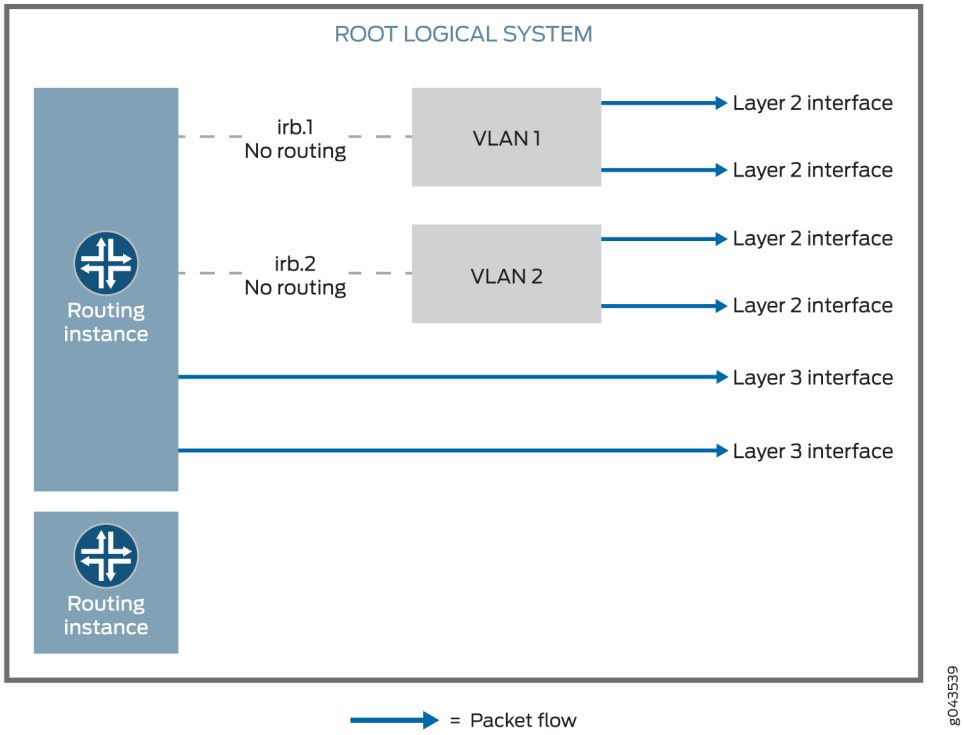
Table 92: Ethernet Physical Interface and Supported Family Types

Ethernet Physical Interface Type	Supported Family Type
Layer 2 Interface	**ethernet-switching**
Layer 3 Interface	**inet** and **inet6**

NOTE: Multiple routing instances are supported.

You can configure both the pseudointerface **irb.x** and the Layer 3 interface under the same default routing instance using either a default routing instance or a user-defined routing instance. See Figure 42 on page 641.

Figure 42: Mixed Transparent and Route Mode



Packets from the Layer 2 interface are switched within the same VLAN, or they connect to the host through the IRB interface. Packets cannot be routed to another IRB interface or a Layer 3 interface through their own IRB interface.

Packets from the Layer 3 interface are routed to another Layer 3 interface. Packets cannot be routed to a Layer 2 interface through an IRB interface.

Table 93 on page 641 lists the security features that are supported in mixed mode and the features that are not supported in transparent mode for Layer 2 switching.

Table 93: Security Features Supported in Mixed Mode (Transparent and Route Mode)

Mode Type	Supported	Not Supported
Mixed mode	• Application Layer Gateways (ALGs) • Firewall User Authentication (FWAUTH) • Intrusion Detection and Prevention (IDP) • Screen • AppSecure	• Unified Threat Management (UTM)

Table 93: Security Features Supported in Mixed Mode (Transparent and Route Mode) (continued)

Mode Type	Supported	Not Supported
Route mode (Layer 3 interface)	• Network Address Translation (NAT) • VPN	—
Transparent mode (Layer 2 interface)		• Network Address Translation (NAT) • VPN • Unified Threat Management (UTM)

Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, some conditions apply to mixed-mode operations. Note the conditions here:

- On SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices, you cannot configure Ethernet switching and virtual private LAN service (VPLS) using mixed mode (Layer 2 and Layer 3).
- On SRX5400, SRX5600, and SRX5800 devices, you do not have to reboot the device when you configure VLAN.

Release History Table

Release	Description
12.3X48-D10	Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, some conditions apply to mixed-mode operations.

Related Documentation

- Example: Improving Security Services by Configuring an SRX Series Device Using Mixed Mode (Transparent and Route Mode) on page 642
- Understanding Secure Wire on Security Devices on page 679

Example: Improving Security Services by Configuring an SRX Series Device Using Mixed Mode (Transparent and Route Mode)

You can configure an SRX Series device using both transparent mode (Layer 2) and route mode (Layer 3) simultaneously to simplify deployments and to improve security services.

This example shows how to pass the Layer 2 traffic from interface ge-0/0/1.0 to interface ge-0/0/0.0 and Layer 3 traffic from interface ge-0/0/2.0 to interface ge-0/0/3.0.

Requirements

This example uses the following hardware and software components:

- An SRX Series device
- Four PCs

Before you begin:

- Create a separate security zone for Layer 2 and Layer 3 interfaces. See "Understanding Layer 2 Security Zones" on page 651.

Overview

In enterprises where different business groups have either Layer 2 or Layer 3 based security solutions, using a single mixed mode configuration simplifies their deployments. In a mixed mode configuration, you can also provide security services with integrated switching and routing.

In addition, you can configure an SRX Series device in both standalone and chassis cluster mode using mixed mode.

In mixed mode (default mode), you can configure both Layer 2 and Layer 3 interfaces simultaneously using separate security zones.

NOTE: **For the mixed mode configuration, you must reboot the device after you commit the changes. However, for SRX5000 line devices, reboot is not required.**

In this example, first you configure a Layer 2 family type called Ethernet switching to identify Layer 2 interfaces. You set the IP address 10.10.10.1/24 to IRB interface. Then you create zone L2 and add Layer 2 interfaces ge-0/0/1.0 and ge-0/0/0.0 to it.

Next you configure a Layer 3 family type inet to identify Layer 3 interfaces. You set the IP address 192.0.2.1/24 to interface ge-0/0/2.0 and the IP address 192.0.2.3/24 to interface ge-0/0/3. Then you create zone L3 and add Layer 3 interfaces ge-0/0/2.0 and ge-0/0/3.0 to it.

Topology

Figure 43 on page 644 shows a mixed mode topology.

Figure 43: Mixed Mode Topology

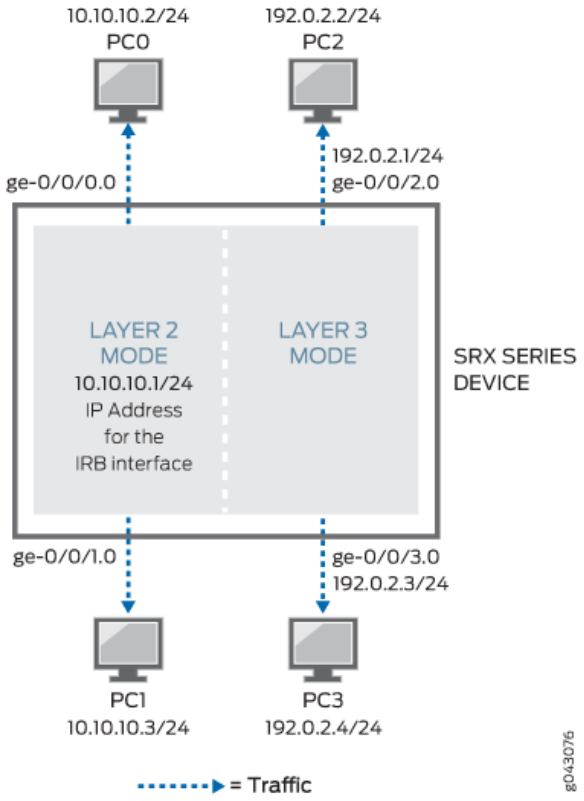


Table 94 on page 644 shows the parameters configured in this example.

Table 94: Layer 2 and Layer 3 Parameters

Parameter	Description
L2	Layer 2 zone.
ge-0/0/1.0 and ge-0/0/0.0	Layer 2 interfaces added to the Layer 2 zone.
L3	Layer 3 zone.
ge-0/0/2.0 and ge-0/0/3.0	Layer 3 interfaces added to the Layer 3 zone.
10.10.10.1/24	IP address for the IRB interface.
192.0.2.1/24 and 192.0.2.3/24	IP addresses for the Layer 3 interface.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 10
set interfaces irb unit 10 family inet address 10.10.10.1/24
set security zones security-zone L2 interfaces ge-0/0/1.0
set security zones security-zone L2 interfaces ge-0/0/0.0
set vlans vlan-10 vlan-id 10
set vlans vlan-10 l3-interface irb.10
set interfaces ge-0/0/2 unit 0 family inet address 192.0.2.1/24
set interfaces ge-0/0/3 unit 0 family inet address 192.0.2.3/24
set security policies default-policy permit-all
set security zones security-zone L2 host-inbound-traffic system-services any-service
set security zones security-zone L2 host-inbound-traffic protocols all
set security zones security-zone L3 host-inbound-traffic system-services any-service
set security zones security-zone L3 host-inbound-traffic protocols all
set security zones security-zone L3 interfaces ge-0/0/2.0
set security zones security-zone L3 interfaces ge-0/0/3.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Layer 2 and Layer 3 interfaces:

1. Create a Layer 2 family type to configure Layer 2 interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/0/0 unit 0 family ethernet-switching interface-mode access
   user@host# set ge-0/0/0 unit 0 family ethernet-switching vlan members 10
   user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
   user@host# set ge-0/0/1 unit 0 family ethernet-switching vlan members 10
   ```

2. Configure an IP address for the IRB interface.

   ```
   [edit interfaces]
   user@host# set irb unit 10 family inet address 10.10.10.1/24
   ```

3. Configure Layer 2 interfaces.

   ```
   [edit security zones security-zone L2 interfaces]
   user@host# set ge-0/0/1.0
   user@host#  set ge-0/0/0.0
   ```

4. Configure VLAN.

```
[edit vlans vlan-10]
user@host# set vlan-id 10
user@host# set l3-interface irb.10
```

5. Configure IP addresses for Layer 3 interfaces.

```
[edit interfaces]
user@host# set ge-0/0/2.0 unit 0 family inet address 192.0.2.1/24
user@host# set ge-0/0/3.0 unit 0 family inet address 192.0.2.3/24
```

6. Configure the policy to permit the traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

7. Configure Layer 3 interfaces.

```
[edit security zones security-zone]
user@host#  set L2 host-inbound-traffic system-services any-service
user@host#  set L2 host-inbound-traffic protocols all
user@host#  set L3 host-inbound-traffic system-services any-service
user@host# set L3 host-inbound-traffic protocols all
user@host#  set L3 interfaces ge-0/0/2.0
user@host# set L3 interfaces ge-0/0/3.0
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show security policies**, **show vlans**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit]
user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan-id 10;
      }
    }
  }
  ge-0/0/1 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan-id 10;
      }
    }
  }
  ge-0/0/2 {
    unit 0 {
      family inet {
        address 192.0.2.1/24;
```



```
        }
      }
    }
    ge-0/0/3 {
      unit 0 {
        family inet {
          address 192.0.2.2/24;
        }
      }
    }
    irb {
      unit 10 {
        family inet {
          address 10.10.10.1/24;
        }
      }
    }
[edit]
user@host#  show security policies
default-policy {
  permit-all;
}
[edit]
user@host# show vlans
  vlan-10 {
    vlan-id 10;
    l3-interface irb.10;
  }
[edit]
user@host#  show security zones
  security-zone L2 {
    host-inbound-traffic {
      system-services {
        any-service;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/1.0;
      ge-0/0/0.0;
    }
  }
  security-zone L3 {
    host-inbound-traffic {
      system-services {
        any-service;
      }
      protocols {
        all;
      }
    }
    interfaces {
      ge-0/0/2.0;
      ge-0/0/3.0;
```

```
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying the Layer 2 and Layer 3 Interfaces and Zones

Purpose Verify that the Layer 2 and Layer 3 interfaces and Layer 2 and Layer 3 zones are created.

Action From operational mode, enter the **show security zones** command.

```
user@host> show security zones
 Security zone: HOST
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 0
  Interfaces:

Security zone: L2
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 2
  Interfaces:
    ge-0/0/0.0
    ge-0/0/1.0

Security zone: L3
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 2
  Interfaces:
    ge-0/0/2.0
    ge-0/0/3.0

Security zone: junos-host
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 0
  Interfaces:
```

Meaning The output shows the Layer 2 (L2) and Layer 3 (L3) zone names and the number and names of Layer 2 and Layer 3 interfaces bound to the L2 and L3 zones.

Verifying the Layer 2 and Layer 3 Session

Purpose Verify that the Layer 2 and Layer 3 sessions are established on the device.

Action From operational mode, enter the **show security flow session** command.

```
user@host> show security flow session
Session ID: 130000050, Policy name: default-policy-00/2, Timeout: 2, Valid
  In: 10.10.10.2/22 --> 10.10.10.3/28;icmp, If: ge-0/0/0.0, Pkts: 1, Bytes: 98
  Out: 10.10.10.3/245 --> 10.10.10.2/248;icmp, If: ge-0/0/1.0, Pkts: 1, Bytes:
98

Session ID: 130000051, Policy name: default-policy-02/2, Timeout: 4, Valid
  In: 192.0.2.1/17 --> 192.0.2.2/19;icmp, If: ge-0/0/2.0, Pkts: 1, Bytes: 84
Out: 192.0.2.2/212 --> 192.0.2.1/218;icmp, If: ge-0/0/3.0, Pkts: 1, Bytes: 84
```

Meaning The output shows active sessions on the device and each session's associated security policy.

- **Session ID 130000050**—Number that identifies the Layer 2 session. Use this ID to get more information about the Layer 2 session such as policy name or number of packets in and out.
- **default-policy-00/2**—Default policy name that permitted the Layer 2 traffic.
- **In**—Incoming flow (source and destination Layer 2 IP addresses with their respective source and destination port numbers, session is ICMP, and the source interface for this session is ge-0/0/0.0).
- **Out**—Reverse flow (source and destination Layer 2 IP addresses with their respective source and destination port numbers, session is ICMP, and destination interface for this session is ge-0/0/1.0).
- **Session ID 130000051**—Number that identifies the Layer 3 session. Use this ID to get more information about the Layer 3 session such as policy name or number of packets in and out.
- **default-policy-02/2**—Default policy name that permitted the Layer 3 traffic.
- **In**—Incoming flow (source and destination Layer 3 IP addresses with their respective source and destination port numbers, session is ICMP, and the source interface for this session is ge-0/0/2.0).
- **Out**—Reverse flow (source and destination Layer 3 IP addresses with their respective source and destination port numbers, session is ICMP, and destination interface for this session is ge-0/0/3.0).

Related Documentation

- Understanding Mixed Mode (Transparent and Route Mode) on Security Devices on page 639
- Understanding Secure Wire on Security Devices on page 679

CHAPTER 25

Configuring Layer 2 Security Zones and Security Policies on Security Devices

Understanding Layer 2 Security Zones

A Layer 2 security zone is a zone that hosts Layer 2 interfaces. A security zone can be either a Layer 2 or Layer 3 zone; it can host either all Layer 2 interfaces or all Layer 3 interfaces, but it cannot contain a mix of Layer 2 and Layer 3 interfaces.

The security zone type—Layer 2 or Layer 3—is implicitly set from the first interface configured for the security zone. Subsequent interfaces configured for the same security zone must be the same type as the first interface.

NOTE: **You cannot configure a device with both Layer 2 and Layer 3 security zones.**

You can configure the following properties for Layer 2 security zones:

- Interfaces—List of interfaces in the zone.
- Policies—Active security policies that enforce rules for the transit traffic, in terms of what traffic can pass through the firewall, and the actions that need to take place on the traffic as it passes through the firewall.
- Screens—A Juniper Networks stateful firewall secures a network by inspecting, and then allowing or denying, all connection attempts that require passage from one security zone to another. For every security zone, and the MGT zone, you can enable a set of predefined screen options that detect and block various kinds of traffic that the device determines as potentially harmful.

NOTE: You can configure the same screen options for a Layer 2 security zone as for a Layer 3 security zone.

- Address books—IP addresses and address sets that make up an address book to identify its members so that you can apply policies to them.
- TCP-RST—When this feature is enabled, the system sends a TCP segment with the reset flag set when traffic arrives that does not match an existing session and does not have the synchronize flag set.

In addition, you can configure a Layer 2 zone for host-inbound traffic. This allows you to specify the kinds of traffic that can reach the device from systems that are directly connected to the interfaces in the zone. You must specify all expected host-inbound traffic because inbound traffic from devices directly connected to the device's interfaces is dropped by default.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Understanding Layer 2 Interfaces on Security Devices on page 637
- Example: Configuring Layer 2 Security Zones on page 652
- Example: Configuring Layer 2 Logical Interfaces on Security Devices on page 638

Example: Configuring Layer 2 Security Zones

This example shows how to configure Layer 2 security zones.

Requirements

Before you begin, determine the properties you want to configure for the Layer 2 security zone. See “Understanding Layer 2 Security Zones” on page 651.

Overview

In this example, you configure security zone l2-zone1 to include a Layer 2 logical interface called ge-3/0/0.0 and security zone l2-zone2 to include a Layer 2 logical interface called ge-3/0/1.0. Then you configure l2-zone2 to allow all supported application services (such as SSH, Telnet, and SNMP) as host-inbound traffic.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network

configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security-zone l2-zone1 interfaces ge-3/0/0.0
set security-zone l2-zone2 interfaces ge-3/0/1.0
set security-zone l2–zone2 host-inbound-traffic system-services all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Layer 2 security zones:

1. Create a Layer 2 security zone and assign interfaces to it.

   ```
   [edit security zones]
   user@host# set security-zone l2-zone1 interfaces ge-3/0/0.0
   user@host# set security-zone l2-zone2 interfaces ge-3/0/1.0
   ```

2. Configure one of the Layer 2 security zones.

   ```
   [edit security zones]
   user@host# set security-zone l2–zone2 host-inbound-traffic system-services all
   ```

3. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Verification

To verify the configuration is working properly, enter the **show security zones** command.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Security Policies in Transparent Mode on page 655
- Example: Configuring Layer 2 Logical Interfaces on Security Devices on page 638

Understanding Security Policies in Transparent Mode

In transparent mode, security policies can be configured only between Layer 2 zones. When packets are forwarded through the VLAN, the security policies are applied between security zones. A security policy for transparent mode is similar to a policy configured for Layer 3 zones, with the following exceptions:

- NAT is not supported.
- IPsec VPN is not supported.

- Application ANY is used.

Layer 2 forwarding does not permit any interzone traffic unless there is a policy explicitly configured on the device. By default, Layer 2 forwarding performs the following actions:

- Allows or denies traffic specified by the configured policy.
- Allows Address Resolution Protocol (ARP) and Layer 2 non-IP multicast and broadcast traffic.
- Continues to block all non-IP and non-ARP unicast traffic.

This default behavior can be changed for Ethernet switching packet flow by using either J-Web or the CLI configuration editor:

- Configure the **block-non-ip-all** option to block all Layer 2 non-IP and non-ARP traffic, including multicast and broadcast traffic.
- Configure the **bypass-non-ip-unicast** option to allow all Layer 2 non-IP traffic to pass through the device.

NOTE: **You cannot configure both options at the same time.**

Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, you can create a separate security zone in mixed mode (the default mode) for Layer 2 and Layer 3 interfaces. However, there is no routing among IRB interfaces and between IRB interfaces and Layer 3 interfaces. Hence, you cannot configure security policies between Layer 2 and Layer 3 zones. You can only configure security policies between the Layer 2 zones or between Layer 3 zones.

Release History Table

Release	Description
12.3X48-D10	Starting in Junos OS Release 12.3X48-D10 and Junos OS Release 17.3R1, you can create a separate security zone in mixed mode (the default mode) for Layer 2 and Layer 3 interfaces.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Security Policies in Transparent Mode on page 655
- Example: Configuring Layer 2 Security Zones on page 652
- Understanding Mixed Mode (Transparent and Route Mode) on Security Devices on page 639

Example: Configuring Security Policies in Transparent Mode

This example shows how to configure security policies in transparent mode between Layer 2 zones.

Requirements

Before you begin, determine the policy behavior you want to include in the Layer 2 security zone. See "Understanding Security Policies in Transparent Mode" on page 653.

Overview

In this example, you configure a security policy to allow HTTP traffic from the 192.0.2.0/24 subnetwork in the l2–zone1 security zone to the server at 192.0.2.1/24 in the l2–zone2 security zone.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 match source-address
  192.0.2.0/24
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 match
  destination-address 192.0.2.1/24
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 match application http
set security policies from-zone l2-zone1 to-zone l2-zone2 policy p1 then permit
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure security policies in transparent mode:

1. Create policies and assign addresses to the interfaces for the zones.

```
[edit security policies]
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 match source-address
  192.0.2.0/24
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 match
  destination-address 192.0.2.1/24
```

2. Set policies for the application.

```
[edit security policies]
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 match application
  http
user@host# set from-zone l2-zone1 to-zone l2-zone2 policy p1 then permit
```

Results From configuration mode, confirm your configuration by entering the **show security policies** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host> show security policies
  from-zone l2-zone1 to-zone l2-zone2
{
  policy p1 {
    match {
    source-address 192.0.2.0/24;
    destination-address 192.0.2.1/24;
    application junos-http;
  }
  then {
    permit;
  }
}
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Layer 2 Security Policies

Purpose Verify that the Layer 2 security policies are configured properly.

Action From configuration mode, enter the **show security policies** command.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Layer 2 Security Zones on page 652

Understanding Firewall User Authentication in Transparent Mode

A firewall user is a network user who must provide a username and password for authentication when initiating a connection across the firewall. Firewall user authentication enables administrators to restrict and permit users accessing protected resources behind a firewall based on their source IP address and other credentials. Junos OS supports the following types of firewall user authentication for transparent mode on the SRX Series device:

- Pass-through authentication—A host or a user from one zone tries to access resources on another zone. You must use an FTP, Telnet, or HTTP client to access the IP address of the protected resource and be authenticated by the firewall. The device uses FTP, Telnet, or HTTP to collect username and password information, and subsequent traffic from the user or host is allowed or denied based on the result of this authentication.
- Web authentication—Users try to connect, by using HTTP, to an IP address on the IRB interface that is enabled for Web authentication. You are prompted for the username and password that are verified by the device. Subsequent traffic from the user or host to the protected resource is allowed or denied based on the result of this authentication.

Related Documentation

- *Authentication and Integrated User Firewalls Feature Guide for Security Devices*
- Layer 2 Transparent Mode Overview on page 377
- Understanding Integrated Routing and Bridging on page 445
- Example: Configuring an IRB Interface on a Security Device on page 452

CHAPTER 26

Configuring IP Spoofing in Layer 2 Transparent Mode on Security Devices

Understanding IP Spoofing in Layer 2 Transparent Mode on Security Devices

In an IP spoofing attack, the attacker gains access to a restricted area of the network and inserts a false source address in the packet header to make the packet appear to come from a trusted source. IP spoofing is most frequently used in denial-of-service (DoS) attacks. When SRX Series devices are operating in transparent mode, the IP spoof-checking mechanism makes use of address book entries. Address books only exist on the Routing Engine. IP spoofing in Layer 2 transparent mode is performed on the Packet Forwarding Engine. Address book information cannot be obtained from the Routing Engine each time a packet is received by the Packet Forwarding Engine. Therefore, address books attached to the Layer 2 zones must be pushed to the Packet Forwarding Engine.

NOTE: **IP spoofing in Layer 2 transparent mode does not support DNS and wildcard addresses.**

When a packet is received by the Packet Forwarding Engine, the packet's source IP address is checked to determine if it is in the incoming zone's address-book. If the packet's source IP address is in the incoming zone's address book, then this IP address is allowed on the interface, and traffic is passed.

If the source IP address is not present in the incoming zone's address-book, but exists in other zones', then the IP address is considered a spoofed IP. Accordingly, actions such as drop and logging can be taken depending on the screen configuration (alarm-without-drop).

NOTE: **If the alarm-without-drop option is configured, the Layer 2 spoofing packet only triggers an alarm message, but the packet is not dropped.**

If a packet's source IP address is not present in the incoming zone's address book or other zones', then you cannot determine if the IP is spoofed or not. In such instances, the packet is passed.

Junos OS takes into account the following match conditions while it searches for source IP addresses in the address book:

- **Host-match**—The IP address match found in the address-book is an address without a prefix.
- **Prefix-match**—The IP address match found in the address-book is an address with a prefix.
- **Any-match**—The IP address match found in the address-book is "any", "any-IPv4", or "any-IPv6".
- **No-match**—No IP address match is found.

Configuring IP Spoofing in Layer 2 Transparent Mode on Security Devices

You can configure the IP spoof-checking mechanism to determine whether or not an IP is being spoofed.

To configure IP spoofing in Layer 2 transparent mode:

1. Set the interface in Layer 2 transparent mode.

   ```
   [edit]
   user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching
   ```

2. (Optional) Set the zone in Layer 2 transparent mode.

   ```
   [edit]
   user@host# set security zones security-zone untrust interfaces ge-0/0/1.0
   ```

3. Configure the address book.

   ```
   [edit]
   user@host# set security address-book my-book address myadd1 10.1.1.0/24
   user@host# set security address-book my-book address myadd2 10.1.2.0/24
   ```

4. Apply the address book to the zone.

   ```
   [edit]
   user@host# set security address-book my-book attach zone untrust
   ```

5. Configure screen IP spoofing.

   ```
   [edit]
   user@host# set security screen ids-option my-screen ip spoofing
   ```

6. Apply the screen to the zone.

   ```
   [edit]
   ```

```
user@host# set security zones security-zone untrust screen my-screen
```

7. (Optional) Configure the **alarm-without-drop** option.

```
[edit]
user@host# set security screen ids-option my-screen alarm-without-drop
```

NOTE: If the **alarm-without-drop** option is configured, the Layer 2 spoofing packet only triggers an alarm message, but the packet is not dropped.

CHAPTER 27

Configuring Class of Service in Transparent Mode on Security Devices

Class of Service Functions in Transparent Mode Overview

Devices operating in Layer 2 transparent mode support the following class-of-service (CoS) functions:

- IEEE 802.1p behavior aggregate (BA) classifiers to determine the forwarding treatment for packets entering the device

  NOTE: Only IEEE 802.1p BA classifier types are supported on devices operating in transparent mode.

- Rewrite rules to redefine IEEE 802.1 CoS values in outgoing packets

  NOTE: Rewrite rules that redefine IP precedence CoS values and Differentiated Services Code Point (DSCP) CoS values are not supported on devices operating in transparent mode.

- Shapers to apply rate limiting to an interface
- Schedulers that define the properties of an output queue

You configure BA classifiers and rewrite rules on transparent mode devices in the same way as on devices operating in Layer 3 mode. For transparent mode devices, however,

you apply BA classifiers and rewrite rules only to logical interfaces configured with the **family ethernet-switching** configuration statement.

Related Documentation

- *Class of Service Feature Guide for Security Devices*
- Layer 2 Transparent Mode Overview on page 377
- Understanding BA Traffic Classification on Transparent Mode Security Devices on page 664
- Example: Configuring BA Classifiers on Transparent Mode Security Devices on page 665

Understanding BA Traffic Classification on Transparent Mode Security Devices

A BA classifier checks the header information of an ingress packet. The resulting traffic classification consists of a forwarding class (FC) and packet loss priority (PLP). The FC and PLP associated with a packet specify the CoS behavior of a hop within the system. For example, a hop can place a packet into a priority queue according to its FC, and manage queues by checking the packet's PLP. Junos OS supports up to eight FCs and four PLPs.

NOTE: **MPLS EXP bit-based traffic classification is not supported.**

BA classification can be applied within one DiffServ domain. BA classification can also be applied between two domains, where each domain honors the CoS results generated by the other domain. Junos OS performs BA classification for a packet by examining its Layer 2 and Layer 3 CoS-related parameters. Those parameters include the following:

- Layer 2—IEEE 802.1p: User Priority
- Layer 3—IPv4 Precedence, IPv4 DSCP, IPv6 DSCP

On SRX Series devices in transparent mode, a BA classifier evaluates only Layer 2 parameters. On SRX Series devices in Layer 3 mode, a BA classifier can evaluate Layer 2 and Layer 3 parameters; in that case, classification resulting from Layer 3 parameters overrides that of Layer 2 parameters.

On SRX Series devices in transparent mode, you specify one of four PLP levels—high, medium-high, medium-low, or low—when configuring a BA classifier.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Class of Service Functions in Transparent Mode Overview on page 663
- Example: Configuring BA Classifiers on Transparent Mode Security Devices on page 665

Example: Configuring BA Classifiers on Transparent Mode Security Devices

This example shows how to configure BA classifiers on transparent mode devices to determine the forwarding treatment of packets entering the devices.

Requirements

Before you begin, configure a Layer 2 logical interface. See "Example: Configuring Layer 2 Logical Interfaces on Security Devices" on page 638.

Overview

In this example, you configure logical interface ge-0/0/4.0 as a trunk port that carries traffic for packets tagged with VLAN identifiers 200 through 390. You then configure forwarding classes and create BA classifier c1 for IEEE 802.1 traffic where incoming packets with IEEE 802.1p priority bits 110 are assigned to the forwarding class fc1 with a low loss priority. Finally, you apply the BA classifier c1 to interface ge-0/0/4.0.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/4 vlan-tagging unit 0 family ethernet-switching interface-mode
    trunk vlan members 200-390
set class-of-service forwarding-classes queue 0 fc1
set class-of-service forwarding-classes queue 1 fc2
set class-of-service forwarding-classes queue 3 fc4
set class-of-service forwarding-classes queue 4 fc5
set class-of-service forwarding-classes queue 5 fc6
set class-of-service forwarding-classes queue 6 fc7
set class-of-service forwarding-classes queue 7 fc8
set class-of-service forwarding-classes queue 2 fc3
set class-of-service classifiers ieee-802.1 c1 forwarding-class fc1 loss-priority low
    code-point 110
set class-of-service interfaces ge-0/0/4 unit 0 classifiers ieee-802.1 c1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure BA classifiers on transparent mode devices:

1. Configure the logical interface as a Layer 2 trunk port.

```
[edit]
user@host# set interfaces ge-0/0/4 vlan-tagging unit 0 family ethernet-switching
  interface-mode trunk vlan members 200-390
```

2. Configure the class of service.

```
[edit]
user@host# edit class-of-service
```

3. Configure the forwarding classes.

```
[edit class-of-service]
user@host# set forwarding-classes queue 0 fc1
user@host# set forwarding-classes queue 1 fc2
user@host# set forwarding-classes queue 3 fc4
user@host# set forwarding-classes queue 4 fc5
user@host# set forwarding-classes queue 5 fc6
user@host# set forwarding-classes queue 6 fc7
user@host# set forwarding-classes queue 7 fc8
user@host# set forwarding-classes queue 2 fc3
```

4. Configure a BA classifier.

```
[edit class-of-service]
user@host# set classifiers ieee-802.1 c1 forwarding-class fc1 loss-priority low
  code-points 110
```

5. Apply the BA classifier to the interface.

```
[edit class-of-service]
user@host# set interfaces ge-0/0/4 unit 0 classifiers ieee-802.1 c1
```

Results From configuration mode, confirm your configuration by entering the **show interfaces ge-0/0/4** and **show class-of-service** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host> show interfaces ge-0/0/4
  vlan-tagging;
  unit 0 {
    family ethernet-switching {
    interface-mode trunk;
    vlan members 200-390;
  }
}
[edit]
  user@host> show class-of-service
  classifiers {
  ieee-802.1 c1 {
    forwarding-class fc1 {
    loss-priority low code-points 110;
    }
```

```
    }
    }
    forwarding-classes {
      queue 0 fc1;
      queue 1 fc2;
      queue 3 fc4;
      queue 4 fc5;
      queue 5 fc6;
      queue 6 fc7;
      queue 7 fc8;
      queue 2 fc3;
      }
      interfaces {
        ge-0/0/4 {
      unit 0 {
      classifiers {
        ieee-802.1 c1;
      }
      }
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

Verifying BA Classifiers on Transparent Mode Devices

Purpose Verify that the BA classifier was configured on the transparent mode devices properly.

Action From configuration mode, enter the **show interfaces ge-0/0/4** and **show class-of-service** commands.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Class of Service Functions in Transparent Mode Overview on page 663
- Understanding BA Traffic Classification on Transparent Mode Security Devices on page 664

Understanding Rewrite of Packet Headers on Transparent Mode Security Devices

Before a packet is transmitted from an interface, the CoS fields in the packet's header can be rewritten for the forwarding class (FC) and packet loss priority (PLP) of the packet. The rewriting function converts a packet's FC and PLP into corresponding CoS fields in the packet header. In Layer 2 transparent mode, the CoS fields are the IEEE 802.1p priority bits.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring Rewrite Rules on Transparent Mode Security Devices on page 668

Example: Configuring Rewrite Rules on Transparent Mode Security Devices

This example shows how to configure rewrite rules on transparent mode devices to redefine IEEE 802.1 CoS values in outgoing packets.

Requirements

Before you begin, configure a Layer 2 logical interface. See "Example: Configuring Layer 2 Logical Interfaces on Security Devices" on page 638.

Overview

In this example, you configure logical interface ge-1/0/3.0 as a trunk port that carries traffic for packets tagged with VLAN identifiers 200 through 390. You then configure the forwarding classes and create rewrite rule rw1 for IEEE 802.1 traffic. For outgoing packets in the forwarding class fc1 with low loss priority, the IEEE 802.1p priority bits are rewritten as 011. Finally, you apply the rewrite rule rw1 to interface ge-1/0/3.0.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-1/0/3 vlan-tagging unit 0 family ethernet-switching interface-mode
    trunk vlan members 200-390
set class-of-service forwarding-classes queue 0 fc1
set class-of-service forwarding-classes queue 1 fc2
set class-of-service forwarding-classes queue 3 fc4
set class-of-service forwarding-classes queue 4 fc5
set class-of-service forwarding-classes queue 5 fc6
set class-of-service forwarding-classes queue 6 fc7
set class-of-service forwarding-classes queue 7 fc8
set class-of-service forwarding-classes queue 2 fc3
set class-of-service rewrite-rules ieee-802.1 rw1 forwarding-class fc1 loss-priority low
    code-point 011
set class-of-service interfaces ge-1/0/3 unit 0 rewrite-rules ieee-802.1 rw1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure rewrite rules on transparent mode devices:

1. Configure the logical interface as a Layer 2 trunk port.

```
[edit]
user@host# set interfaces ge-1/0/3 vlan-tagging unit 0 family ethernet-switching
  interface-mode trunk vlan members 200-390
```

2. Configure the class of service.

```
[edit]
user@host# edit class-of-service
```

3. Configure the forwarding classes.

```
[edit class-of-service]
user@host# set forwarding-classes queue 0 fc1
user@host# set forwarding-classes queue 1 fc2
user@host# set forwarding-classes queue 3 fc4
user@host# set forwarding-classes queue 4 fc5
user@host# set forwarding-classes queue 5 fc6
user@host# set forwarding-classes queue 6 fc7
user@host# set forwarding-classes queue 7 fc8
user@host# set forwarding-classes queue 2 fc3
```

4. Configure a rewrite rule.

```
[edit class-of-service]
user@host# set rewrite-rules ieee-802.1 rw1 forwarding-class fc1 loss-priority low
  code-point 011
```

5. Apply the rewrite rule to the interface.

```
[edit class-of-service]
user@host# set interfaces ge-1/0/3 unit 0 rewrite-rules ieee-802.1 rw1
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces ge-1/0/3** and **show class-of-service** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host> show interfaces ge-1/0/3
  vlan-tagging;
  unit 0 {
    family ethernet-switching {
    interface-mode trunk;
    vlan members 200-390;
  }
}
```

```
[edit]
  user@host> show class-of-service
  forwarding-classes {
  queue 0 fc1;
    queue 1 fc2;
    queue 3 fc4;
    queue 4 fc5;
    queue 5 fc6;
    queue 6 fc7;
    queue 7 fc8;
  queue 2 fc3;
  }
  interfaces {
    ge-1/0/3 {
    unit 0 {
    rewrite-rules {
      ieee-802.1 rw1;
      }
    }
  }
  }
  rewrite-rules {
    ieee-802.1 rw1 {
    forwarding-class fc1 {
    loss-priority low code-point 011;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform this task:

Verifying Rewrite Rules on Transparent Mode Devices

Purpose Verify that the rewrite rule was configured on the transparent mode devices properly.

Action From configuration mode, enter the **show interfaces ge-1/0/3** and **show class-of-service** commands.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Understanding Rewrite of Packet Headers on Transparent Mode Security Devices on page 667

CHAPTER 28

Configuring IPv6 Flows on Security Devices

Understanding IPv6 Flows in Transparent Mode on Security Devices

In transparent mode, the SRX Series device filters packets that traverse the device without modifying any of the source or destination information in the packet MAC headers. Transparent mode is useful for protecting servers that mainly receive traffic from untrusted sources because there is no need to reconfigure the IP settings of routers or protected servers.

A device operates in transparent mode when all physical interfaces on the device are configured as Layer 2 interfaces. A physical interface is a Layer 2 interface if its logical interface is configured with the **ethernet-switching** option at the [**edit interfaces** ***interface-name*** **unit** ***unit-number*** **family**] hierarchy level. There is no command to define or enable transparent mode on the device. The device operates in transparent mode when there are interfaces defined as Layer 2 interfaces. The device operates in route mode (the default mode) if all physical interfaces are configured as Layer 3 interfaces.

By default, IPv6 flows are dropped on security devices. To enable processing by security features such as zones, screens, and firewall policies, you must enable flow-based forwarding for IPv6 traffic with the **mode flow-based** configuration option at the [**edit security forwarding-options family inet6**] hierarchy level. You must reboot the device when you change the mode.

In transparent mode, you can configure Layer 2 zones to host Layer 2 interfaces, and you can define security policies between Layer 2 zones. When packets travel between Layer 2 zones, security policies can be enforced on these packets. The following security features are supported for IPv6 traffic in transparent mode:

- Layer 2 security zones and security policies. See "Understanding Layer 2 Security Zones" on page 651 and "Understanding Security Policies in Transparent Mode" on page 653.
- Firewall user authentication. See "Understanding Firewall User Authentication in Transparent Mode" on page 656.

- Layer 2 transparent mode chassis clusters. See *Understanding Layer 2 Transparent Mode Chassis Clusters on Security Devices*.
- Class of service functions. See "Class of Service Functions in Transparent Mode Overview" on page 663.

The following security features are *not* supported for IPv6 flows in transparent mode:

- Logical systems
- IPv6 GTPv2
- J-Web interface
- NAT
- IPsec VPN
- With the exception of DNS, FTP, and TFTP ALGs, all other ALGs are not supported.

Configuring VLANs and Layer 2 logical interfaces for IPv6 flows is the same as configuring VLANs and Layer 2 logical interfaces for IPv4 flows. You can optionally configure an integrated routing and bridging (IRB) interface for management traffic in a VLAN. The IRB interface is the only Layer 3 interface allowed in transparent mode. The IRB interface on the SRX Series device does not support traffic forwarding or routing. The IRB interface can be configured with both IPv4 and IPv6 addresses. You can assign an IPv6 address for the IRB interface with the **address** configuration statement at the [**edit interfaces irb unit *number* family inet6**] hierarchy level. You can assign an IPv4 address for the IRB interface with the **address** configuration statement at the [**edit interfaces irb unit *number* family inet**] hierarchy level.

The Ethernet Switching functions on SRX Series devices are similar to the switching features on Juniper Networks MX Series routers. However, not all Layer 2 networking features supported on MX Series routers are supported on SRX Series devices. See "Layer 2 Transparent Mode Overview" on page 377.

The SRX Series device maintains forwarding tables that contain MAC addresses and associated interfaces for each Layer 2 VLAN. The IPv6 flow processing is similar to IPv4 flows. See "Layer 2 Learning and Forwarding for VLANs Overview" on page 27.

Related Documentation

- Flow-Based Processing for IPv6 Traffic on Security Devices on page 672
- Example: Configuring Transparent Mode for IPv6 Flows on Security Devices on page 674

Flow-Based Processing for IPv6 Traffic on Security Devices

Flow-based processing mode is required for security features such as zones, screens, and firewall policies to function. By default, the SRX Series device is enabled for flow-based forwarding for IPv6 traffic on all devices, apart from the SRX300 Series and SRX550M devices that are set to drop mode. Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, for the SRX1500 series, SRX4100, SRX4200, SRX5400, SRX5600, SRX5800 and vSRX devices, you do *not* need to reboot the device when you are switching modes between flow mode, packet mode, and drop mode. For SRX300

Series and SRX550M devices, you *must* reboot the device when switching between flow mode, packet mode, and drop mode.

SRX300 Series and the SRX550M Devices

When IPv6 is configured on SRX300 Series and the SRX550M devices, the default behavior is set to drop mode because of memory constraints. In this case, you must reboot the device after changing the processing mode from the drop mode default to flow-based processing mode or packet-based processing mode—that is, between modes on these devices.

NOTE: For drop mode processing, the traffic is dropped directly, it is not forwarded. It differs from packet-mode processing for which the traffic is handled but no security processes are applied.

To process IPv6 traffic on SRX300 Series and the SRX550M devices, you need to configure IPv6 addresses for the transit interfaces that receive and forward the traffic. For information about the inet6 protocol family and procedures for configuring IPv6 addresses for interfaces, see *Interfaces Feature Guide for Security Devices*.

Configuring an SRX Series Device as a Border Router

When an SRX Series device of any type is enabled for flow-based processing or drop mode, to configure the device as a border router you must change the mode to packet-based processing for MPLS. In this case, to configure the SRX device to packet mode for MPLS, use the **set security forwarding-options family mpls mode packet-based** statement.

NOTE: As mentioned, for SRX300 Series and the SRX550M devices, whenever you change processing modes, you must reboot the device.

Enabling Flow-Based Processing for IPv6 Traffic on SRX300 Series and SRX550M Devices

To enable flow-based forwarding for IPv6 traffic on SRX300 Series and the SRX550M devices, modify the mode at the [**edit security forwarding-options family inet6**] hierarchy level:

```
security {
  forwarding-options {
    family {
      inet6 {
        mode flow-based;
      }
    }
  }
}
```

To configure forwarding for IPv6 traffic on SRX300 Series or an SRX500M device:

1. Change the forwarding option mode for IPv6 to flow-based.

```
[edit]
user@host# security forwarding-options family inet6 mode flow-based
```

2. Review your configuration.

```
[edit]
user@host# show security forwarding-options
  family {
    inet6 {
      mode flow-based;
    }
  }
```

3. Commit the configuration.

```
[edit]
user@host# commit
```

4. Reboot the device.

NOTE: For SRX300 Series and SRX500M devices, the device discards IPv6 type 0 Routing Header (RH0) packets.

Release History Table

Release	Description
15.1X49-D70	By default, the SRX Series device is enabled for flow-based forwarding for IPv6 traffic on all devices, apart from the SRX300 Series and SRX550M devices that are set to drop mode. Starting with Junos OS Release 15.1X49-D70 and Junos OS Release 17.3R1, for the SRX1500 series, SRX4100, SRX4200, SRX5400, SRX5600, SRX5800 and vSRX devices, you do *not* need to reboot the device when you are switching modes between flow mode, packet mode, and drop mode. For SRX300 Series and SRX550M devices, you *must* reboot the device when switching between flow mode, packet mode, and drop mode.

Example: Configuring Transparent Mode for IPv6 Flows on Security Devices

This example shows how to configure VLANs, a Layer 2 interface, and an IRB interface that supports both IPv4 and IPv6 addresses. This example also shows how to configure the device to use only ARP requests to learn the outgoing interfaces for unknown destination MAC addresses.

Requirements

The device must be enabled for IPv6 flow processing. See "Flow-Based Processing for IPv6 Traffic on Security Devices" on page 672.

Overview

This example creates the configuration described in Table 95 on page 675.

Table 95: IPv6 Transparent Mode Configuration for IPv6 Flows

Feature	Name	Configuration Parameters
VLANs	vlan-a	VLAN 2
	vlan-b	VLAN 10
Logical interface	ge-0/0/0.0	Trunk port for packets tagged with VLAN IDs 1 through 10
Physical interface	ge-0/0/0	VLAN ID 30 assigned to untagged packets
IRB interface	irb.0	Addresses: • IPv4 address 10.1.1.1/24 • IPv6 address 2001:0db8:2::1/64 Referenced in vlan-b VLAN
Learn the outgoing interfaces for unknown destination MAC addresses		Use only ARP queries without traceroute requests

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-a vlan-id 2
set vlans vlan-b vlan members 1-10
set interfaces ge-0/0/0 vlan-tagging native-vlan-id 30
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode trunk vlan
  members 1-10
set interfaces irb unit 0 family inet address 10.1.1.1/24
set interfaces irb unit 0 family inet6 address 2001:0db8::1/64
set vlans vlan-b l3-interface irb.0
set security flow ethernet-switching no-packet-flooding no-trace-route
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure transparent mode for IPv6 flows:

1. Configure VLANs.

```
[edit vlans]
user@host# set vlan-a vlan-id 2
user@host# set vlan-b vlan members 1-10
```

2. Configure the Layer 2 interface.

```
[edit interfaces ge-0/0/0]
user@host# set vlan-tagging native-vlan-id 30
user@host# set unit 0 family ethernet-switching interface-mode trunk vlan members
  1-10
```

3. Configure the IRB interface.

```
[edit interfaces irb unit 0]
user@host# set family inet address 10.1.1.1/24
user@host# set family inet6 address 2001:0db8::1/64
```

4. Configure the IRB interface for the VLAN.

```
[edit vlans]
user@host# set vlan-b l3-interface irb.0
```

5. Configure learning for unknown destination MAC addresses.

```
[edit security flow ethernet-switching]
user@host# set no-packet-flooding no-trace-route
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, and **show security flow ethernet-switching** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show vlans
vlan-a {
  vlan-id 2;
}
  vlan-b {
    vlan members 1-10;
    l3-interface irb.0;
  }
user@host# show interfaces
ge-0/0/0 {
  vlan-tagging;
```

```
    native-vlan-id 30;
    unit 0 {
      family ethernet-switching {
        interface-mode trunk;
        vlan members 1-10;
      }
    }
  }
  user@host# show security flow ethernet-switching
  no-packet-flooding {
    no-trace-route;
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying IPv6 Sessions on page 677
- Verifying IPv6 Gates on page 677
- Verifying IPv6 IP-action Settings on page 677

Verifying IPv6 Sessions

Purpose Verify IPv6 sessions on the device.

Action From operational mode, enter the **show security flow session family inet6** command.

Verifying IPv6 Gates

Purpose Verify IPv6 gates on the device.

Action From operational mode, enter the **show security flow gate family inet6** command.

Verifying IPv6 IP-action Settings

Purpose Verify IPv6 IP-action settings on the device.

Action From operational mode, enter the **show security flow ip-action family inet6** command.

Related Documentation

- *Understanding IPv6 Address Space, Addressing, Address Format, and Address Types*

- Understanding IPv6 Flows in Transparent Mode on Security Devices on page 671

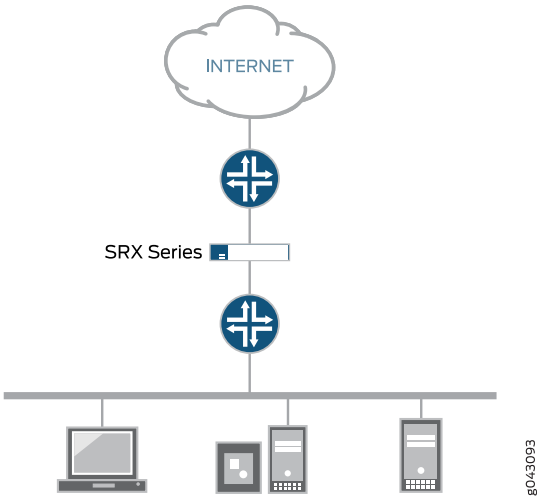
CHAPTER 29

Configuring Secure Wire on Security Devices

Understanding Secure Wire on Security Devices

Traffic that arrives on a specific interface can be forwarded unchanged through another interface. This mapping of interfaces, called secure wire, allows an SRX Series to be deployed in the path of network traffic without requiring a change to routing tables or a reconfiguration of neighboring devices. Figure 44 on page 680 shows a typical in-path deployment of an SRX Series with secure wire.

Figure 44: SRX Series In-Path Deployment with Secure Wire



Secure wire maps two peer interfaces. It differs from transparent and route modes in that there is no switching or routing lookup to forward traffic. As long as the traffic is permitted by a security policy, a packet arriving on one peer interface is immediately forwarded unchanged out of the other peer interface. There is no routing or switching decision made on the packet. Return traffic is also forwarded unchanged.

Secure wire mapping is configured with the **secure-wire** statement at the [**edit security forwarding-options**] hierarchy level; two Ethernet logical interfaces must be specified. The Ethernet logical interfaces must be configured with **family ethernet-switching** and each pair of interfaces must belong to the VLAN(s). The interfaces must be bound to security zones and a security policy configured to permit traffic between the zones.

This feature is available on Ethernet logical interfaces only; both IPv4 and IPv6 traffic are supported. You can configure interfaces for access or trunk mode. Secure wire supports chassis cluster redundant Ethernet interfaces. This feature does not support security features not supported in transparent mode, including NAT and IPsec VPN. Layer 7 features, including AppSecure, and IPS/IDP are supported.

NOTE: **Layer 7 feature, UTM is not supported in secure wire.**

Secure wire is a special case of Layer 2 transparent mode on SRX Series devices that provide point-to-point connections. This means that the two interfaces of a secure wire must ideally be directly connected to Layer 3 entities, such as routers or hosts. Secure wire interfaces can be connected to switches. However, note that a secure wire interface forwards all arriving traffic to the peer interface only if the traffic is permitted by a security policy.

Secure wire can coexist with Layer 3 mode. While you can configure Layer 2 and Layer 3 interfaces at the same time, traffic forwarding occurs independently on Layer 2 and Layer 3 interfaces.

Secure wire can coexist with Layer 2 transparent mode. If both features exist on the same SRX Series device, you need to configure them in different VLANs.

NOTE: Integrated routing and bridging (IRB) interfaces are not supported with secure wire.

Related Documentation

- Example: Simplifying SRX Series Device Deployment with Secure Wire over Access Mode Interfaces on page 681
- Example: Simplifying SRX Series Device Deployment with Secure Wire over Trunk Mode Interfaces on page 685
- Example: Simplifying SRX Series Device Deployment with Secure Wire over Aggregated Interface Member Links on page 688
- Example: Simplifying Chassis Cluster Deployment with Secure Wire over Redundant Ethernet Interfaces on page 693
- Example: Simplifying Chassis Cluster Deployment with Secure Wire over Aggregated Redundant Ethernet Interfaces on page 698
- Understanding Mixed Mode (Transparent and Route Mode) on Security Devices on page 639

Example: Simplifying SRX Series Device Deployment with Secure Wire over Access Mode Interfaces

If you are connecting an SRX Series device to other network devices, you can use secure wire to simplify the device deployment in the network. No changes to routing or forwarding tables on the SRX Series device and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified access mode interfaces on an SRX Series device as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series device to other network devices through access mode interfaces.

This example shows how to configure a secure wire mapping for two access mode interfaces. This configuration applies to scenarios where user traffic is not VLAN tagged.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures the secure wire access-sw that maps interface ge-0/0/0.0 to interface ge-0/0/1.0. The two peer interfaces are configured for access mode. The VLAN ID 10 is configured for the vlan-10 and the access mode interfaces.

NOTE: A specific VLAN ID must be configured for a VLAN.

Topology

Figure 45 on page 682 shows the access mode interfaces that are mapped in secure wire access-sw.

Figure 45: Secure Wire Access Mode Interfaces

Configuration

CLI Quick Configuration

NOTE: Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, some Layer 2 CLI configuration statements are enhanced, and some commands are changed. For detailed information about the modified hierarchies, see "Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices" on page 385.

The configuration statements shown below are for Junos OS Release 15.1X49-D10 or higher and Junos OS Release 17.3R1.

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-10 vlan-id 10
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members 10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 10
set security forwarding-options secure-wire access-sw interface [ge-0/0/0.0 ge-0/0/1.0]
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for access mode interfaces:

1. Configure the VLAN.

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

2. Configure the access mode interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/0 unit 0 family ethernet-switching vlan members 10
user@host# set ge-0/0/1 unit 0 family ethernet-switching vlan members 10
```

3. Configure the secure wire mapping.

```
[edit security forwarding-options]
user@host# set secure-wire access-sw interface [ge-0/0/0.0 ge-0/0/1.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/0.0
user@host# set security-zone untrust interfaces ge-0/0/1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
  vlan-10 {
    vlan-id 10;
    interface ge-0/0/0.0;
    interface ge-0/0/1.0;
  }
  user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan {
          members 10;
```

```
        }
      }
    }
  }
  ge-0/0/1 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan {
          members 10;
        }
      }
    }
  }
  user@host# show security forwarding-options
  secure-wire {
    access-sw {
      interface [ ge-0/0/0.0 ge-0/0/1.0 ];
    }
  }
user@host# show security zones
  security-zone trust {
    interfaces {
      ge-0/0/0.0;
    }
  }
  security-zone untrust {
    interfaces {
      ge-0/0/1.0;
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying Secure Wire Mapping on page 684
- Verifying the VLAN on page 685

Verifying Secure Wire Mapping

Purpose Verify the secure wire mapping.

Action From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
Secure wire                    Interface       Link   Interface       Link

access-sw                      ge-0/0/0.0      up     ge-0/0/1.0      up
Total secure wires: 1
```

Verifying the VLAN

Purpose Verify the VLAN.

Action From operational mode, enter the **show vlans vlan-10** command.

```
user@host> show vlans vlan-10
Routing instance        VLAN name             Tag     Interfaces
default-switch          vlan-10                10         ge-0/0/0.0
                                                          ge-0/0/1.0
```

Related Documentation
- Understanding Secure Wire on Security Devices on page 679

Example: Simplifying SRX Series Device Deployment with Secure Wire over Trunk Mode Interfaces

If you are connecting an SRX Series device to other network devices, you can use secure wire to simplify the device deployment in the network. No changes to routing or forwarding tables on the SRX Series device and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified trunk mode interfaces on an SRX Series device as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series device to other network devices through trunk mode interfaces.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures the secure wire trunk-sw that maps interface ge-0/1/0.0 to interface ge-0/1/1.0. The two peer interfaces are configured for trunk mode and carry user traffic tagged with VLAN IDs from 100 to 102. The VLAN ID list 100-102 is configured for the VLAN vlan-100 and the trunk mode interfaces.

NOTE: A specific VLAN ID must be configured for a VLAN.

Topology

Figure 46 on page 686 shows the trunk mode interfaces that are mapped in secure wire trunk-sw.

Figure 46: Secure Wire Trunk Mode Interfaces

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-100 vlan members 100-102
set interfaces ge-0/1/0 unit 0 family ethernet-switching interface-mode trunk vlan
  members 100-102
set interfaces ge-0/1/1 unit 0 family ethernet-switching interface-mode trunk vlan
  members 100-102
set security forwarding-options secure-wire trunk-sw interface [ge-0/1/0.0 ge-0/1/1.0]
set security zones security-zone trust interfaces ge-0/1/0.0
set security zones security-zone untrust interfaces ge-0/1/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for trunk mode interfaces:

1. Configure the VLAN.

   ```
   [edit vlans vlan-100]
   user@host# set vlan members 100-102
   ```

2. Configure the trunk mode interfaces.

   ```
   [edit interfaces]
   user@host# set ge-0/1/0 unit 0 family ethernet-switching interface-mode trunk
     vlan members 100-102
   user@host# set ge-0/1/1 unit 0 family ethernet-switching interface-mode trunk
     vlan members 100-102
   ```

3. Configure the secure wire mapping.

   ```
   [edit security forwarding-options]
   user@host# set secure-wire trunk-sw interface [ge-0/1/0.0 ge-0/1/1.0]
   ```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/1/0.0
user@host# set security-zone untrust interfaces ge-0/1/1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
  vlan-100 {
    vlan members 100-102;
  }
  user@host# show interfaces
  ge-0/1/0 {
    unit 0 {
      family ethernet-switching {
        interface-mode trunk;
        vlan members 100-102;
      }
    }
  }
  ge-0/1/1 {
    unit 0 {
      family ethernet-switching {
        interface-mode trunk;
        vlan members 100-102;
      }
    }
  }
  user@host# show security forwarding-options
  secure-wire trunk-sw {
    interfaces [ge-0/1/0.0 ge-0/1/1.0];
  }
user@host# show security zones
  security-zone trust {
    interfaces {
      ge-0/1/0.0;
    }
  }
  security-zone untrust {
    interfaces {
      ge-0/1/1.0;
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying Secure Wire Mapping

Purpose Verify the secure wire mapping.

Action From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
Secure wire                    Interface       Link   Interface       Link

trunk-sw                       ge-0/1/0.0      up     ge-0/1/1.0      up
Total secure wires: 1
```

Verifying the VLAN

Purpose Verify the VLAN.

Action From operational mode, enter the **show vlans** command.

```
user@host> show vlans
Routing instance        VLAN name             VLAN ID     Interfaces
default-switch          vlan-100-vlan-0100          100        ge-0/1/0.0
                                                               ge-0/1/1.0
default-switch          vlan-100-vlan-0101          101        ge-0/1/0.0
                                                               ge-0/1/1.0
default-switch          vlan-100-vlan-0102          102        ge-0/1/0.0
                                                               ge-0/1/1.0
```

NOTE: VLANs are automatically expanded, with one VLAN for each VLAN ID in the VLAN ID list.

Related Documentation

Example: Simplifying SRX Series Device Deployment with Secure Wire over Aggregated Interface Member Links

If you are connecting an SRX Series device to other network devices, you can use secure wire to simplify the device deployment in the network. No changes to routing or forwarding

tables on the SRX Series device and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified aggregated interface member links on an SRX Series device as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series device to other network devices through aggregated interface member links.

NOTE: LACP is not supported. Secure wire mappings can be configured for member links of link bundles instead of directly mapping aggregated Ethernet interfaces.

NOTE: On SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX550, and SRX650 devices, when you create an aggregated interface with two or more ports and set the family to Ethernet switching, and if a link in the bundle goes down, the traffic forwarded through the same link will be rerouted two seconds later. This causes an outage for the traffic being sent to the link until reroute is complete.

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example configures secure wires for two aggregated Ethernet interface link bundles with two links each. Two separate secure wires ae-link1 and ae-link2 are configured using one link from each aggregated Ethernet link bundle. This static mapping requires that the two link bundles have the same number of links.

For link bundles, all logical interfaces of the secure wire mappings must belong to the same VLAN. VLAN ID 10 is configured for the VLAN vlan-10 and the logical interfaces. All logical interfaces of a link bundle must belong to the same security zone.

NOTE: A specific VLAN ID or VLAN ID list must be configured for a VLAN.

Topology

Figure 47 on page 689 shows the aggregated interfaces that are mapped in secure wire configurations.

Figure 47: Secure Wire Aggregated Interfaces

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-10 vlan-id 10
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access vlan-id
  10
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access vlan-id
  10
set interfaces ge-0/1/0 unit 0 family ethernet-switching interface-mode access vlan-id
  10
set interfaces ge-0/1/1 unit 0 family ethernet-switching interface-mode access vlan-id
  10
set security forwarding-options secure-wire ae-link1-sw interface [ge-0/1/0.0 ge-0/1/1.0]
set security forwarding-options secure-wire ae-link2-sw interface [ge-0/0/0.0 ge-0/0/1.0]
set security zones security-zone trust interfaces ge-0/0/0.0
set security zones security-zone trust interfaces ge-0/1/0.0
set security zones security-zone untrust interfaces ge-0/0/1.0
set security zones security-zone untrust interfaces ge-0/1/1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for aggregated interface member links:

1. Configure the VLAN.

   ```
   [edit vlans vlan-10]
   user@host# set vlan-id10
   ```

2. Configure the interfaces.

   ```
   [edit interfaces ]
   user@host# set ge-0/0/0 unit 0 family ethernet-switching interface-mode access
     vlan-id 10
   user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
     vlan-id 10
   user@host# set ge-0/1/0 unit 0 family ethernet-switching interface-mode access
     vlan-id 10
   user@host# set ge-0/1/1 unit 0 family ethernet-switching interface-mode access
     vlan-id 10
   ```

3. Configure the secure wire mappings.

   ```
   [edit security forwarding-options]
   user@host# set secure-wire ae-link1-sw interface [ ge-0/1/0.0 ge-0/1/1.0 ]
   user@host# set secure-wire ae-link2-sw interface [ ge-0/0/0.0 ge-0/0/1.0 ]
   ```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces ge-0/0/0.0
user@host# set security-zone trust interfaces ge-0/1/0.0
user@host# set security-zone untrust interfaces ge-0/0/1.0
user@host# set security-zone untrust interfaces ge-0/1/1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
  vlan-10 {
    vlan-id 10;
  }
  user@host# show interfaces
  ge-0/0/0 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan-id 10;
      }
    }
  }
  ge-0/0/1 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan-id 10;
      }
    }
  }
  ge-0/1/0 {
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan-id 10;
      }
    }
  }
  ge-0/1/1{
    unit 0 {
      family ethernet-switching {
        interface-mode access;
        vlan-id 10;
      }
    }
```

```
    }
    user@host# show security forwarding-options
    secure-wire ae-link1-sw {
      interfaces [ge-0/1/0.0 ge-0/1/1.0];
    }
    secure-wire ae-link2-sw {
      interfaces [ge-0/0/0.0 ge-0/0/1.0];
    }
  user@host# show security zones
    security-zone trust {
      interfaces {
        ge-0/0/0.0;
        ge-0/1/0.0;
      }
    }
    security-zone untrust {
      interfaces {
        ge-0/0/1.0;
        ge-0/1/1.0;
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying Secure Wire Mapping on page 692
- Verifying the VLAN on page 692

Verifying Secure Wire Mapping

Purpose Verify the secure wire mapping.

Action From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
Secure wire                    Interface       Link   Interface       Link

ae-link1-sw                    ge-0/1/0.0      up     ge-0/1/1.0      up
ae-link2-sw                    ge-0/0/0.0      up     ge-0/0/1.0      up
Total secure wires: 2
```

Verifying the VLAN

Purpose Verify the VLAN.

Action From operational mode, enter the **show vlans vlan-10** command.

```
user@host> show vlans vlan-10
```

```
Routing instance        VLAN name            VLAN ID     Interfaces
default-switch          vlan-10                   10          ge-0/0/0.0
                                                              ge-0/0/1.0
                                                              ge-0/1/0.0
                                                              ge-0/1/1.0
```

Related Documentation

- Understanding Secure Wire on Security Devices on page 679

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Redundant Ethernet Interfaces

If you are connecting an SRX Series chassis cluster to other network devices, you can use secure wire to simplify the cluster deployment in the network. No changes to routing or forwarding tables on the cluster and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified redundant Ethernet interfaces on the SRX Series chassis cluster as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series chassis cluster to other network devices through redundant Ethernet interfaces.

Requirements

Before you begin:

- Connect a pair of the same SRX Series devices in a chassis cluster.
- Configure the chassis cluster node ID and cluster ID.
- Set the number of redundant Ethernet interfaces in the chassis cluster.
- Configure the chassis cluster fabric.
- Configure chassis cluster redundancy group (in this example redundancy group 1 is used).

For more information, see the *Chassis Cluster Feature Guide for SRX Series Devices*.

Overview

Secure wire is supported over redundant Ethernet interfaces in a chassis cluster. The two redundant Ethernet interfaces must be configured in the same redundancy group. If failover occurs, both redundant Ethernet interfaces must fail over together.

NOTE: Secure wire mapping of redundant Ethernet link aggregation groups (LAGs) are not supported. LACP is not supported.

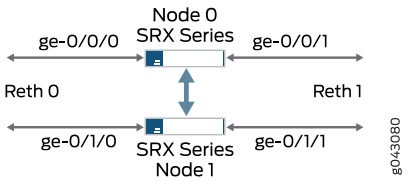
This example configures the secure wire reth-sw that maps ingress interface reth0.0 to egress interface reth1.0. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster. The two redundant Ethernet interfaces are configured for access mode. VLAN ID 10 is configured for the VLAN vlan-10 and the redundant Ethernet interfaces.

NOTE: A specific VLAN ID or VLAN ID list must be configured for a VLAN.

Topology

Figure 48 on page 694 shows the redundant Ethernet interfaces that are mapped in secure wire reth-sw.

Figure 48: Secure Wire Redundant Ethernet Interfaces



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-10 vlan-id 10
set interfaces ge-0/0/0 gigether-options redundant-parent reth0
set interfaces ge-0/0/1 gigether-options redundant-parent reth1
set interfaces ge-0/1/0 gigether-options redundant-parent reth0
set interfaces ge-0/1/1 gigether-options redundant-parent reth1
set interfaces reth0 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth1 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1
set security forwarding-options secure-wire reth-sw interface [reth0.0 reth1.0]
set security zones security-zone trust interfaces reth0.0
set security zones security-zone untrust interfaces reth1.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for chassis cluster redundant Ethernet interfaces:

1. Configure the VLAN.

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

2. Configure the redundant Ethernet interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/0 gigether-options redundant-parent reth0
user@host# set ge-0/0/1 gigether-options redundant-parent reth1
user@host# set ge-0/1/0 gigether-options redundant-parent reth0
user@host# set ge-0/1/1 gigether-options redundant-parent reth1

user@host#set reth0 unit 0 family ethernet-switching interface-mode access vlan-id
  10
user@host#set reth1 unit 0 family ethernet-switching interface-mode access vlan-id
  10

user@host# set reth0 redundant-ether-options redundancy-group 1
user@host# set reth1 redundant-ether-options redundancy-group 1
```

3. Configure the secure wire mapping.

```
[edit security forwarding-options]
user@host# set secure-wire reth-sw interface [reth0.0 reth1.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces reth0.0
user@host# set security-zone untrust interfaces reth1.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results

From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
  vlan-10 {
    vlan-id 10;
  }
```

```
user@host# show interfaces
ge-0/0/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-0/0/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
ge-0/1/0 {
    gigether-options {
        redundant-parent reth0;
    }
}
ge-0/1/1 {
    gigether-options {
        redundant-parent reth1;
    }
}
reth0 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan-id 10;
        }
    }
}
reth1 {
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 0 {
        family ethernet-switching {
            interface-mode access;
            vlan-id 10;
        }
    }
}
user@host# show security forwarding-options
secure-wire reth-sw {
    interfaces [reth0.0 reth1.0];
}
user@host# show security zones
security-zone trust {
    interfaces {
        reth0.0;
    }
}
security-zone untrust {
    interfaces {
        reth1.0;
```

```
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying Secure Wire Mapping on page 697
- Verifying the VLAN on page 697

Verifying Secure Wire Mapping

Purpose Verify the secure wire mapping.

Action From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
node0:
--------------------------------------------------------------------------
Secure wire                        Interface       Link   Interface       Link

reth-sw                            reth0.0         up     reth1.0         up

Total secure wires: 1

node1:
--------------------------------------------------------------------------
Secure wire                        Interface       Link   Interface       Link

reth-sw                            reth0.0         up     reth1.0         up

Total secure wires: 1
```

Verifying the VLAN

Purpose Verify the VLAN.

Action From operational mode, enter the **show vlan vlan-10** command.

```
user@host> show vlan vlan-10
Routing instance        VLAN Name            VLAN ID     Interfaces
default-switch          vlan-10                   10         reth0.0
                                                             reth1.0
```

Related Documentation

- Understanding Secure Wire on Security Devices on page 679
- Example: Simplifying Chassis Cluster Deployment with Secure Wire over Aggregated Redundant Ethernet Interfaces on page 698

Example: Simplifying Chassis Cluster Deployment with Secure Wire over Aggregated Redundant Ethernet Interfaces

If you are connecting an SRX Series chassis cluster to other network devices, you can use secure wire to simplify the cluster deployment in the network. No changes to routing or forwarding tables on the cluster and no reconfiguration of neighboring devices is needed. Secure wire allows traffic to be forwarded unchanged between specified redundant Ethernet interfaces on the SRX Series chassis cluster as long as it is permitted by security policies or other security features. Follow this example if you are connecting an SRX Series chassis cluster to other network devices through aggregated redundant Ethernet interfaces.

NOTE: Secure wires cannot be configured for redundant Ethernet interface link aggregation groups (LAGs). For the secure wire mapping shown in this example, there is no LAG configuration on the SRX Series chassis cluster. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster. Users on upstream or downstream devices connected to the SRX Series cluster can configure the redundant Ethernet interface child links in LAGs.

Requirements

Before you begin:

- Connect a pair of the same SRX Series devices in a chassis cluster.
- Configure the chassis cluster node ID and cluster ID.
- Set the number of redundant Ethernet interfaces in the chassis cluster.
- Configure the chassis cluster fabric.
- Configure the chassis cluster redundancy group (in this example, redundancy group 1 is used).

For more information, see the *Chassis Cluster Feature Guide for SRX Series Devices*.

Overview

This example configures secure wires for four redundant Ethernet interfaces: reth0, reth1, reth2, and reth3. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster. All four redundant Ethernet interfaces must be in the same VLAN—in this example, the VLAN is vlan-0. Two of the redundant Ethernet

interfaces, reth0.0 and reth2.0, are assigned to the trust zone, while the other two interfaces, reth1.0 and reth3.0, are assigned to the untrust zone.

This example configures the following secure wires:

- reth-sw1 maps interface reth0.0 to interface reth1.0
- reth-sw2 maps interface reth2.0 to reth3.0

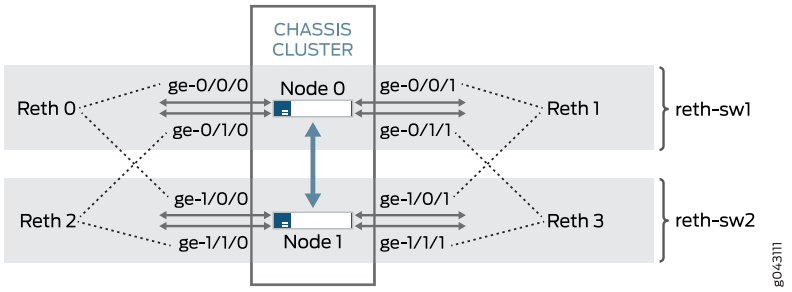
All redundant Ethernet interfaces are configured for access mode. VLAN ID 10 is configured for the VLAN vlan-0 and the redundant Ethernet interfaces.

NOTE: A specific VLAN ID or VLAN ID list must be configured for a VLAN.

Topology

Figure 49 on page 699 shows the redundant Ethernet interface child links that are mapped in secure wire configurations reth-sw1 and reth-sw2. Each redundant Ethernet interface consists of two child interfaces, one on each node of the chassis cluster.

Figure 49: Secure Wire Redundant Ethernet Interface Child Links



Users on upstream or downstream devices connected to the SRX Series cluster can configure redundant Ethernet interface child links in a LAG as long as the LAG does not span chassis cluster nodes. For example, ge-0/0/0 and ge-0/1/0 and ge-0/0/1 and ge-0/1/1 on node 0 can be configured as LAGs on connected devices. In the same way, ge-1/0/0 and ge-1/1/0 and ge-1/0/1 and ge-1/1/1 on node 1 can be configured as LAGs on connected devices.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans vlan-0 vlan-id 10
set interfaces ge-0/0/0 gigether-options redundant-parent reth0
set interfaces ge-0/0/1 gigether-options redundant-parent reth1
set interfaces ge-0/1/0 gigether-options redundant-parent reth2
```

```
set interfaces ge-0/1/1 gigether-options redundant-parent reth3
set interfaces ge-1/0/0 gigether-options redundant-parent reth0
set interfaces ge-1/0/1 gigether-options redundant-parent reth1
set interfaces ge-1/1/0 gigether-options redundant-parent reth2
set interfaces ge-1/1/1 gigether-options redundant-parent reth3
set interfaces reth0 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth1 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth2 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth3 unit 0 family ethernet-switching interface-mode access vlan-id 10
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth2 redundant-ether-options redundancy-group 1
set interfaces reth3 redundant-ether-options redundancy-group 1
set security forwarding-options secure-wire reth-sw1 interface [reth0.0 reth1.0]
set security forwarding-options secure-wire reth-sw2 interface [reth2.0 reth3.0]
set security zones security-zone trust interfaces reth0.0
set security zones security-zone trust interfaces reth2.0
set security zones security-zone untrust interfaces reth1.0
set security zones security-zone untrust interfaces reth3.0
set security policies default-policy permit-all
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a secure wire mapping for aggregated interface member links:

1. Configure the VLAN.

```
[edit vlans vlan-0]
user@host# set vlan-id 10
```

2. Configure the redundant Ethernet interfaces.

```
[edit interfaces ]
user@host# set ge-0/0/0 gigether-options redundant-parent reth0
user@host# set ge-0/0/1 gigether-options redundant-parent reth1
user@host# set ge-0/1/0 gigether-options redundant-parent reth2
user@host# set ge-0/1/1 gigether-options redundant-parent reth3
user@host# set ge-1/0/0 gigether-options redundant-parent reth0
user@host# set ge-1/0/1 gigether-options redundant-parent reth1
user@host# set ge-1/1/0 gigether-options redundant-parent reth2
user@host# set ge-1/1/1 gigether-options redundant-parent reth3

user@host# set reth0 unit 0 family ethernet-switching interface-mode access
  vlan-id 10
user@host# set reth1 unit 0 family ethernet-switching interface-mode access vlan-id
  10
user@host# set reth2 unit 0 family ethernet-switching interface-mode access vlan-id
  10
user@host# set reth3 unit 0 family ethernet-switching interface-mode access vlan-id
  10

user@host# set reth0 redundant-ether-options redundancy-group 1
```

```
user@host# set reth1 redundant-ether-options redundancy-group 1
user@host# set reth2 redundant-ether-options redundancy-group 1
user@host# set reth3 redundant-ether-options redundancy-group 1
```

3. Configure the secure wire mappings.

```
[edit security forwarding-options]
user@host# set secure-wire reth-sw1 interface [reth0.0 reth1.0]
user@host# set secure-wire reth-sw2 interface [reth2.0 reth3.0]
```

4. Configure security zones.

```
[edit security zones]
user@host# set security-zone trust interfaces reth0.0
user@host# set security-zone trust interfaces reth2.0

user@host# set security-zone untrust interfaces reth1.0
user@host# set security-zone untrust interfaces reth3.0
```

5. Configure a security policy to permit traffic.

```
[edit security policies]
user@host# set default-policy permit-all
```

Results From configuration mode, confirm your configuration by entering the **show vlans**, **show interfaces**, **show security forwarding-options**, and **show security zones** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show vlans
  vlan-0 {
    vlan-id 10;
  }
  user@host# show interfaces
  ge-0/0/0 {
    gigether-options {
      redundant-parent reth0;
    }
  }
  ge-0/0/1 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  ge-0/1/0 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  ge-0/1/1 {
    gigether-options {
      redundant-parent reth3;
    }
```



```
}
ge-1/0/0 {
  gigether-options {
    redundant-parent reth0;
  }
}
ge-1/0/1 {
  gigether-options {
    redundant-parent reth1;
  }
}
ge-1/1/0 {
  gigether-options {
    redundant-parent reth2;
  }
}
ge-1/1/1 {
  gigether-options {
    redundant-parent reth3;
  }
}
reth0 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan-id 10;
    }
  }
}
reth1 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan-id 10;
    }
  }
}
reth2 {
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 0 {
    family ethernet-switching {
      interface-mode access;
      vlan-id 10;
    }
  }
}
reth3 {
  redundant-ether-options {
```

```
        redundancy-group 1;
      }
      unit 0 {
        family ethernet-switching {
          interface-mode access;
          vlan-id 10;
        }
      }
    }
    user@host# show security forwarding-options
    secure-wire reth-sw1 {
      interfaces [reth0.0 reth1.0];
    }
    secure-wire reth-sw2 {
      interfaces [reth2.0 reth3.0];
    }
  user@host# show security zones
    security-zone trust {
      interfaces {
        reth0.0;
        reth2.0;
      }
    }
    security-zone untrust {
      interfaces {
        reth1.0;
        reth3.0;
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying Secure Wire Mapping on page 703
- Verifying VLAN on page 704

Verifying Secure Wire Mapping

Purpose Verify the secure wire mapping.

Action From operational mode, enter the **show security forwarding-options secure-wire** command.

```
user@host> show security forward-options secure-wire
node0:
--------------------------------------------------------------------------
Secure wire                       Interface       Link   Interface       Link

reth-sw1                          reth0.0         up     reth1.0         up
reth-sw2                          reth2.0         up     reth3.0         up

Total secure wires: 2
```

```
node1:
--------------------------------------------------------------------------
Secure wire                         Interface       Link   Interface       Link

reth-sw1                            reth0.0         up     reth1.0         up
reth-sw2                            reth2.0         up     reth3.0         up

Total secure wires: 2
```

Verifying VLAN

Purpose Verify the VLAN.

Action From operational mode, enter the **show vlans vlan-0** command.

```
user@host> show vlans vlan-0
Routing instance        VLAN name             VLAN ID     Interfaces
default-switch          vlan-0                    10
                                                            reth0.0
                                                            reth1.0
                                                            reth2.0
                                                            reth3.0
```

Related Documentation

- Understanding Secure Wire on Security Devices on page 679
- Example: Simplifying Chassis Cluster Deployment with Secure Wire over Redundant Ethernet Interfaces on page 693

CHAPTER 30

Configuring Ethernet Port Switching Modes on Security Devices

Understanding Switching Modes on Security Devices

There are two types of switching modes:

- Switching Mode—The uPIM appears in the list of interfaces as a single interface, which is the first interface on the uPIM. For example, ge-2/0/0. You can optionally configure each uPIM port only for autonegotiation, speed, and duplex mode. A uPIM in switching mode can perform the following functions:
  - Layer 3 forwarding—Routes traffic destined for WAN interfaces and other PIMs present on the chassis.
  - Layer 2 forwarding—Switches intra-LAN traffic from one host on the LAN to another LAN host (one port of uPIM to another port of same uPIM).
- Enhanced Switching Mode—Each port can be configured for switching or routing mode. This usage differs from the routing and switching modes, in which all ports must be in either switching or routing mode. The uPIM in enhanced switching mode provides the following features:
  - Supports configuration of different types of VLANs and inter-VLAN routing.
  - Supports Layer 2 control plane protocol such as Link Aggregation Control Protocol (LACP).
  - Supports port-based Network Access Control (PNAC) by means of authentication servers.

NOTE: The SRX300 and SRX320 devices support enhanced switching mode only. When you set a multiport uPIM to enhanced switching mode, all the Layer 2 switching features are supported on the uPIM. (Platform support depends on the Junos OS release in your installation.)

You can set a multiport Gigabit Ethernet uPIM on a device to either switching or enhanced switching mode.

When you set a multiport uPIM to switching mode, the uPIM appears as a single entity for monitoring purposes. The only physical port settings that you can configure are autonegotiation, speed, and duplex mode on each uPIM port, and these settings are optional.

Related Documentation

- Example: Configuring Switching Modes on Security Devices on page 713
- Ethernet Ports Switching Overview for Security Devices on page 706

Ethernet Ports Switching Overview for Security Devices

Certain ports on Juniper Networks devices can function as Ethernet access switches that switch traffic at Layer 2 and route traffic at Layer 3.

You can deploy supported devices in branch offices as an access or desktop switch with integrated routing capability, thus eliminating intermediate access switch devices from your network topology. The Ethernet ports provide switching while the Routing Engine provides routing functionality, enabling you to use a single device to provide routing, access switching, and WAN interfaces.

This topic contains the following sections:

Supported Devices and Ports

Juniper Networks supports switching features on a variety of Ethernet ports and devices (see Table 96 on page 706). Platform support depends on the Junos OS release in your installation. The following ports and devices are included:

- Onboard Ethernet ports (Gigabit and Fast Ethernet built-in ports) on the SRX300, SRX320, SRX320 PoE, SRX340, SRX345, SRX550M and SRX1500 devices.
- Multiport Gigabit Ethernet XPIM on the SRX650 device.

Table 96: Supported Devices and Ports for Switching Features

Device	Ports
SRX100 devices	Onboard Fast Ethernet ports (fe-0/0/0 and fe-0/0/7)

Table 96: Supported Devices and Ports for Switching Features (continued)

Device	Ports
SRX210 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 and ge-0/0/1) and 1-Port Gigabit Ethernet SFP Mini-PIM port. Onboard Fast Ethernet ports (fe-0/0/2 and fe-0/0/7)
SRX220 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/7) and 1-Port Gigabit Ethernet SFP Mini-PIM port.
SRX240 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/15) and 1-Port Gigabit Ethernet SFP Mini-PIM port.
SRX300 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/7)
SRX320 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/7)
SRX340 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/15)
SRX345 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/15)
SRX550 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/9, Multiport Gigabit Ethernet XPIM modules, and 1-Port Gigabit Ethernet SFP Mini-PIM port.
SRX550M devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/9 and Multiport Gigabit Ethernet XPIM modules.
SRX650 devices	Multiport Gigabit Ethernet XPIM modules NOTE: On SRX650 devices, Ethernet switching is not supported on Gigabit Ethernet interfaces (ge-0/0/0 through ge-0/0/3 ports).
SRX1500 devices	Onboard Gigabit Ethernet ports (ge-0/0/0 through ge-0/0/19)

On the SRX100. SRX220, SRX240, SRX300, SRX320, SRX340 and SRX345 devices, you can set the onboard Gigabit Ethernet ports to operate as either switched ports or routed ports. (Platform support depends on the Junos OS release in your installation.)

Integrated Bridging and Routing

Integrated bridging and routing (IRB) provides support for simultaneous Layer 2 swiching and Layer 3 routing within the same VLAN. Packets arriving on an interface of the VLAN are switched or routed based on the destination MAC address of the packet. Packets with the router's MAC address as the destination are routed to other Layer 3 interfaces.

Link Layer Discovery Protocol and LLDP-Media Endpoint Discovery

Devices use Link Layer Discovery Protocol (LLDP) and LLDP-Media Endpoint Discovery (MED) to learn and distribute device information about network links. The information

allows the device to quickly identify a variety of systems, resulting in a LAN that interoperates smoothly and efficiently.

LLDP-capable devices transmit information in Type Length Value (TLV) messages to neighbor devices. Device information can include specifics, such as chassis and port identification and system name and system capabilities. The TLVs leverage this information from parameters that have already been configured in the Junos OS.

LLDP-MED goes one step further, exchanging IP-telephony messages between the device and the IP telephone. These TLV messages provide detailed information about Power over Ethernet (PoE) policy. The PoE Management TLVs let the device ports advertise the power level and power priority needed. For example, the device can compare the power needed by an IP telephone running on a PoE interface with available resources. If the device cannot meet the resources required by the IP telephone, the device could negotiate with the telephone until a compromise on power is reached.

The following basic TLVs are supported:

- Chassis Identifier—The MAC address associated with the local system.
- Port identifier—The port identification for the specified port in the local system.
- Port Description—The user-configured port description. The port description can be a maximum of 256 characters.
- System Name—The user-configured name of the local system. The system name can be a maximum of 256 characters.
- Switching Features Overview—This information is not configurable, but taken from the software.
- System Capabilities—The primary function performed by the system. The capabilities that system supports; for example, Ethernet switching or router. This information is not configurable, but based on the model of the product.
- Management Address—The IP management address of the local system.

The following LLDP-MED TLVs are supported:

- LLDP-MED Capabilities—A TLV that advertises the primary function of the port. The values range from 0 through 15:
  - 0—Capabilities
  - 1—Network policy
  - 2—Location identification
  - 3—Extended power through medium-dependent interface power-sourcing equipment (MDI-PSE)
  - 4—Inventory
  - 5–15—Reserved
- LLDP-MED Device Class Values:

- 0—Class not defined
- 1—Class 1 device
- 2—Class 2 device
- 3—Class 3 device
- 4—Network connectivity device
- 5–255— Reserved

NOTE: Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, Link Layer Discovery Protocol (LLDP) and LLDP-Media Endpoint Discovery (MFD) are enabled on SRX300, SRX320, SRX340, SRX345, SRX550M and SRX1500 devices.

- Network Policy—A TLV that advertises the port VLAN configuration and associated Layer 2 and Layer 3 attributes. Attributes include the policy identifier, application types, such as voice or streaming video, 802.1Q VLAN tagging, and 802.1p priority bits and Diffserv code points.
- Endpoint Location—A TLV that advertises the physical location of the endpoint.
- Extended Power via MDI—A TLV that advertises the power type, power source, power priority, and power value of the port. It is the responsibility of the PSE device (network connectivity device) to advertise the power priority on a port.

LLDP and LLDP-MED must be explicitly configured on uPIMs (in enhanced switching mode) on base ports on SRX100, SRX210, SRX240, SRX300, SRX320, SRX340, and SRX345 devices, and Gigabit Backplane Physical Interface Modules (GPIMs) on SRX650 devices. (Platform support depends on the Junos OS release in your installation.) To configure LLDP on all interfaces or on a specific interface, use the **lldp** statement at the [**set protocols**] hierarchy level. To configure LLDP-MED on all interfaces or on a specific interface, use the **lldp-med** statement at the [**set protocols**] hierarchy level.

Types of Switch Ports

The ports, or interfaces, on a switch operate in either access mode or trunk mode.

An interface in access mode connects to a network device, such as a desktop computer, an IP telephone, a printer, a file server, or a security camera. The interface itself belongs to a single VLAN. The frames transmitted over an access interface are normal Ethernet frames.

Trunk interfaces 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 one another.

uPIM in a Daisy Chain

You cannot combine multiple uPIMs to act as a single integrated switch. However, you can connect uPIMs on the same chassis externally by physically connecting a port on one uPIM to a port on another uPIM in a daisy-chain fashion.

Two or more uPIMs daisy-chained together create a single switch with a higher port count than either individual uPIM. One port on each uPIM is used solely for the connection. For example, if you daisy-chain a 6-port uPIM and an 8-port uPIM, the result operates as a 12-port uPIM. Any port of a uPIM can be used for daisy chaining.

Configure the IP address for only one of the daisy-chained uPIMs, making it the primary uPIM. The secondary uPIM routes traffic to the primary uPIM, which forwards it to the Routing Engine. This results in some increase in latency and packet drops due to oversubscription of the external link.

Only one link between the two uPIMs is supported. Connecting more than one link between uPIMs creates a loop topology, which is not supported.

Q-in-Q VLAN Tagging

Q-in-Q tunneling, defined by the IEEE 802.1ad standard, allows service providers on Ethernet access networks to extend a Layer 2 Ethernet connection between two customer sites.

In Q-in-Q tunneling, as a packet travels from a customer VLAN (C-VLAN) to a service provider's VLAN, a service provider-specific 802.1Q tag is added to the packet. This additional tag is used to segregate traffic into service-provider-defined service VLANs (S-VLANs). The original customer 802.1Q tag of the packet remains and is transmitted transparently, passing through the service provider's network. As the packet leaves the S-VLAN in the downstream direction, the extra 802.1Q tag is removed.

NOTE: When Q-in-Q tunneling is configured for a service provider's VLAN, all Routing Engine packets, including packets from the routed VLAN interface, that are transmitted from the customer-facing access port of that VLAN will always be untagged.

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

- All-in-one bundling—Use the **dot1q-tunneling** statement at the [**edit vlans**] hierarchy level to map without specifying customer VLANs. All packets from a specific access interface are mapped to the S-VLAN.
- Many-to-one bundling—Use the **customer-vlans** statement at the [**edit vlans**] hierarchy level to specify which C-VLANs are mapped to the S-VLAN.
- Mapping C-VLAN on a specific interface—Use the **mapping** statement at the [**edit vlans**] hierarchy level to map a specific C-VLAN on a specified access interface to the S-VLAN.

Table 97 on page 711 lists the C-VLAN to S-VLAN mapping supported on SRX Series devices. (Platform support depends on the Junos OS release in your installation.)

Table 97: Supported Mapping Methods

Mapping	SRX210	SRX240	SRX300	SRX320	SRX340	SRX345	SRX550M	SRX650
All-in-one bundling	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Many-to-one bundling	No	No	No	No	Yes	Yes	Yes	Yes
Mapping C-VLAN on a specific interface	No	No	No	No	Yes	Yes	Yes	Yes

NOTE: VLAN translation is supported on SRX300 and SRX320 devices and these devices do not support Q-in-Q tunneling.

NOTE: On SRX650 devices, in the dot1q-tunneling configuration options, customer VLANs range and VLAN push do not work together for the same S-VLAN, even when you commit the configuration. If both are configured, then VLAN push takes priority over customer VLANs range.

IRB interfaces are supported on Q-in-Q VLANs for SRX210, SRX240, SRX340, SRX345, and SRX650 devices. Packets arriving on an IRB interface on a Q-in-Q VLAN are routed regardless of whether the packet is single or double tagged. The outgoing routed packets contain an S-VLAN tag only when exiting a trunk interface; the packets exit the interface untagged when exiting an access interface. (Platform support depends on the Junos OS release in your installation.)

In a Q-in-Q deployment, customer packets from downstream interfaces are transported without any changes to source and destination MAC addresses. You can disable MAC address learning at both the interface level and the VLAN level. Disabling MAC address learning on an interface disables learning for all the VLANs of which that interface is a member. When you disable MAC address learning on a VLAN, MAC addresses that have already been learned are flushed.

On SRX100, SRX210, SRX240, SRX300, SRX320, SRX340, SRX345, and SRX650 devices (with platform support depending on the Junos OS release in your installation), on the Layer 3 aggregated Ethernet, the following features are not supported:

- Encapsulations (such as CCC, VLAN CCC, VPLS, and PPPoE)
- J-Web

- On all SRX Series devices, the Link Layer Discovery Protocol (LLDP) is not supported on redundant Ethernet (reth) interfaces.
- On SRX550M devices the aggregate Ethernet (ae) interface with XE member interface cannot be configured with the Ethernet switching family.
- On SRX300, SRX320, SRX340, SRX345, and SRX550M devices, the Q-in-Q support on a Layer 3 interface has the following limitations:
  - Double tagging is not supported on reth and ae interfaces.
  - Multitopology routing is not supported in flow mode and in chassis clusters.
  - Dual tagged frames are not supported on encapsulations (such as CCC, TCC, VPLS, and PPPoE)
  - On Layer 3 logical interfaces, **input-vlan-map**, **output-vlan-map**, **inner-range**, and **inner-list** are not applicable
  - Only TPIDs with 0x8100 are supported, and the maximum number of tags is 2.
  - Dual tagged frames are accepted only for logical interfaces with IPV4 and IPv6 families.
- On SRX100, SRX210, SRX240, SRX300, SRX320, SRX340, SRX345, and SRX650 devices (with platform support depending on the Junos OS release in your installation), on the routed VLAN interface (RVI), the following features are not supported:
  - IS-IS (family ISO)
  - Encapsulations (Ether CCC, VLAN CCC, VPLS, PPPoE, and so on) on VLAN interfaces
  - CLNS
  - DVMRP
  - VLAN interface MAC change
  - G-ARP
  - Change VLAN-Id for VLAN interface

Release History Table

Release	Description
15.1X49-D60	Starting in Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, Link Layer Discovery Protocol (LLDP) and LLDP-Media Endpoint Discovery (MFD) are enabled on SRX300, SRX320, SRX340, SRX345, SRX550M and SRX1500 devices.

Related Documentation

- Understanding Switching Modes on Security Devices on page 705

Example: Configuring Switching Modes on Security Devices

Requirements

Before you begin, see "Ethernet Ports Switching Overview for Security Devices" on page 706.

Overview

In this example, you configure **chassis** and set the l2-learning protocol to global mode switching. You then set a physical port parameter on the l2-learning protocols.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols l2-learning global-mode switching
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
```

Step-by-Step Procedure

To configure switching mode:

1. Set l2-learning protocol to global mode switching.

   ```
   [edit protocols l2-learning]
   user@host# set protocols l2-learning global-mode switching
   ```

2. Set a physical port parameter on the l2-learning protocols.

   ```
   [edit]
   user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode
     access
   ```

3. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results

From configuration mode, confirm your configuration by entering the **show protocols** and **show interfaces** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show protocols
```

```
l2-learning {
  global-mode switching;
}
```

```
[edit]
user@host# show interfaces
ge-0/0/1 {
  unit 0 {
    family ethernet-switching {
      interface-mode access;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying the Switching Mode on page 714
- Verifying the Ethernet switching on Interface ge-0/0/1 on page 715

Verifying the Switching Mode

Purpose Make sure that the switching mode is configured as expected.

Action From operational mode, enter the **show ethernet-switching global-information** command.

```
user@host> show ethernet-switching global-information
```

```
Global Configuration:

MAC aging interval    : 300
MAC learning          : Enabled
MAC statistics        : Disabled
MAC limit Count       : 16383
MAC limit hit         : Disabled
MAC packet action drop: Disabled
MAC+IP aging interval : IPv4 - 1200 seconds
                        IPv6 - 1200 seconds
MAC+IP limit Count    : 393215
MAC+IP limit reached  : No
LE  aging time        : 1200
LE  VLAN aging time   : 1200
Global Mode           : Switching
RE state              : Master
```

Meaning The sample output shows that the global mode switching is configured as expected.

Verifying the Ethernet switching on Interface ge-0/0/1

Purpose Make sure that the Ethernet switching is configured as expected on interface ge-0/0/1.

Action From operational mode, enter the **show interfaces ge-0/0/1 brief** command.

```
user@host> show interfaces ge-0/0/1 brief
```

```
Physical interface: ge-0/0/1, Enabled, Physical link is Down
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 1000mbps, Loopback:
 Disabled, Source filtering: Disabled, Flow control: Disabled, Auto-negotiation:
 Enabled, Remote fault: Online
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None

  Logical interface ge-0/0/1.0
    Flags: Device-Down SNMP-Traps 0x0 Encapsulation: Ethernet-Bridge
    Security: Zone: Null
    eth-switch
```

Meaning The sample output shows that the Ethernet switching is configured on interface ge-0/0/1 as expected .

Related Documentation

- Ethernet Ports Switching Overview for Security Devices on page 706

CHAPTER 31

Configuring Class of Service in Switching Mode on Security Devices

Class of Service Functions in Switching Mode Overview

When a network experiences congestion and delay, some packets must be dropped. Juniper Networks Junos operating system (Junos OS) class of service (CoS) divides traffic into classes to which you can apply different levels of throughput and packet loss when congestion occurs. This allows packet loss to happen according to rules that you configure.

For interfaces that carry IPv4, IPv6, and MPLS traffic, you can configure Junos OS CoS features to provide multiple classes of service for different applications. CoS also allows you to rewrite the Differentiated Services code point (DSCP), IP precedence, 802.1p, or EXP CoS bits of packets egressing an interface, thus allowing you to tailor packets for the remote peers' network requirements.

CoS provides multiple classes of service for different applications. You can configure multiple forwarding classes for transmitting packets, define which packets are placed into each output queue, and schedule the transmission service level for each queue.

In designing CoS applications, you must carefully consider your service needs and thoroughly plan and design your CoS configuration to ensure consistency and interoperability across all platforms in a CoS domain.

Related Documentation

- Understanding Junos OS CoS Components for SRX Series Devices on page 718

Understanding Junos OS CoS Components for SRX Series Devices

This topic describes the Juniper Networks Junos OS class-of-service (CoS) components for Juniper Networks SRX Series devices:

Code-Point Aliases

A code-point alias assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other CoS components such as classifiers, drop-profile maps, and rewrite rules.

Policers

Policers limit traffic of a certain class to a specified bandwidth and *burst size*. Packets exceeding the policer limits can be discarded. You define policers with filters that can be associated with input interfaces.

NOTE: You can configure policers to discard packets that exceed the rate limits. If you want to configure CoS parameters such as loss-priority and forwarding-class, you must use firewall filters.

Classifiers

Packet classification associates incoming packets with a particular CoS servicing level. In Junos OS, *classifiers* associate packets with a forwarding class and loss priority and assign packets to output queues on the basis of associated forwarding classes. Junos OS supports two general types of classifiers:

- Behavior aggregate or CoS value traffic classifiers—Examines the CoS value in the packet header. The value in this single field determines the CoS settings applied to the packet. BA classifiers allow you to set the forwarding class and loss priority of a packet based on the Differentiated Services code point (DSCP) value, IP precedence value, and IEEE 802.1p value.
- Multifield traffic classifiers—Examines multiple fields in the packet such as source and destination addresses and source and destination port numbers of the packet. With multifield classifiers, you set the forwarding class and loss priority of a packet on the basis of firewall filter rules.

Forwarding Classes

Forwarding classes group the packets for transmission. Based on forwarding classes, you assign packets to output queues. Forwarding classes affect the forwarding, scheduling, and marking policies applied to packets as they transit a device. By default, four categories of forwarding classes are defined: best effort, assured forwarding, expedited forwarding, and network control. SRX Series devices support, 16 forwarding classes, providing granular classification capability.

Tail Drop Profiles

Drop profile is a mechanism that defines parameters that enable packets to be dropped from the network. Drop profiles define the meanings of the loss priorities. When you configure drop profiles, you are essentially setting the value for queue fullness. The queue fullness represents a percentage of the queue used to store packets in relation to the total amount that has been allocated for that specific queue.

Loss priorities set the priority of dropping a packet. Loss priority affects the scheduling of a packet without affecting the packet's relative ordering. You can use the loss priority setting to identify packets that have experienced congestion. Typically, you mark packets exceeding some service level with a high loss priority.

Schedulers

Each switch interface has multiple queues assigned to store packets. The switch determines which queue to service with regard to a particular method of scheduling. This process often involves determining which type of packet must be transmitted before another. You can define the priority, bandwidth, delay buffer size, and tail drop profiles to be applied to a particular queue for packet transmission.

A scheduler map associates a specified forwarding class with a scheduler configuration. You can associate up to four user-defined scheduler maps with the interfaces.

Rewrite Rules

A *rewrite rule* sets the appropriate CoS bits in the outgoing packet, thus allowing the next downstream device to classify the packet into the appropriate service group. Rewriting, or marking, outbound packets is useful when the switch is at the border of a network and must alter the CoS values to meet the policies of the targeted peer.

NOTE: Egress firewall filters can also assign forwarding class and loss priority so that the packets are rewritten based on forwarding class and loss priority.

Related Documentation

- Class of Service Functions in Switching Mode Overview on page 718

Classification Overview

Packet classification refers to the examination of an incoming packet, which associates the packet with a particular class-of-service (CoS) servicing level. Junos operating system (OS) supports these classifiers:

- Behavior aggregate (BA) classifiers
- Multifield (MF) classifiers
- Default IP precedence classifiers

NOTE: The total number of classifiers supported on a Services Processing Unit (SPU) is 79. Three classifiers are installed on the SPU as default classifiers in the Layer 3 mode, independent of any CoS configuration, which leaves 76 classifiers that can be configured using the CoS CLI commands. The default classifiers number can vary in future releases or in different modes.

Verify the number of default classifiers installed on the SPU to determine how many classifiers can be configured using the CoS CLI commands.

When both BA and MF classifications are performed on a packet, the MF classification has higher precedence.

In Junos OS, classifiers associate incoming packets with a forwarding class (FC) and packet loss priority (PLP), and, based on the associated FC, assign packets to output queues. A packet's FC and PLP specify the behavior of a hop, within the system, to process the packet. The per-hop behavior (PHB) comprises packet forwarding, policing, scheduling, shaping, and marking. For example, a hop can put a packet in one of the priority queues according to its FC and then manage the queues by checking the packet's PLP. Junos OS supports up to eight FCs and four PLPs.

This topic includes the following sections:

Behavior Aggregate Classifiers

A BA classifier operates on a packet as it enters the device. Using BA classifiers, the device aggregates different types of traffic into a single FC so that all the types of traffic will receive the same forwarding treatment. The CoS value in the packet header is the single field that determines the CoS settings applied to the packet. BA classifiers allow you to set a packet's FC and PLP based on the Differentiated Services (DiffServ) code point (DSCP) value, DSCP IPv4 value, DSCP IPv6 value, IP precedence value, MPLS EXP bits, or IEEE 802.1p value. The default classifier is based on the IP precedence value. For more information, see "Default IP Precedence Classifier" on page 723.

Junos OS performs BA classification for a packet by examining its Layer 2, Layer 3, and related CoS parameters, as shown in Table 98 on page 721.

Table 98: BA Classification

Layer	CoS Parameter
Layer 2	IEEE 802.1p value: User Priority
Layer 3	IPv4 precedence IPv4 Differentiated Services code point (DSCP) value IPv6 DSCP value

NOTE: A BA classifier evaluates Layer 2 and Layer 3 parameters independently. The results from Layer 2 parameters override the results from the Layer 3 parameters.

Multifield Classifiers

An MF classifier is a second means of classifying traffic flows. Unlike the BA classifier, an MF classifier can examine multiple fields in the packet—for example, the source and destination address of the packet, or the source and destination port numbers of the packet. With MF classifiers, you set the FC and PLP based on firewall filter rules.

NOTE: For a specified interface, you can configure both an MF classifier and a BA classifier without conflicts. Because the classifiers are always applied in sequential order (the BA classifier followed by the MF classifier) any BA classification result is overridden by an MF classifier if they conflict.

Junos OS performs MF traffic classification by directly scrutinizing multiple fields of a packet to classify a packet. This avoids having to rely on the output of the previous BA traffic classification. Junos OS can simultaneously check a packet's data for Layers 2, 3, 4, and 7, as shown in Table 99 on page 722.

Table 99: MF Classification

Layer	CoS Parameter
Layer 2	IEEE 802.1Q: VLAN ID IEEE 802.1p: User priority
Layer 3	IP precedence value DSCP or DSCP IPv6 value Source IP address Destination IP address Protocol ICMP: Code and type
Layer 4	TCP/UDP: Source port TCP/UDP: Destination port TCP: Flags AH/ESP: SPI
Layer 7	Not supported.

Using Junos OS, you configure an MF classifier with a firewall filter and its associated match conditions. This enables you to use any filter match criterion to locate packets that require classification.

Default IP Precedence Classifier

With Junos OS, all logical interface are automatically assigned a default IP precedence classifier when the logical interface is configured. This default traffic classifier maps IP precedence values to an FC and a PLP as shown in Table 100 on page 723. These mapping results are in effect for an ingress packet until the packet is further processed by another classification method.

Table 100: Default IP Precedence Classifier

IP Precedence CoS Values	Forwarding Class	Packet Loss Priority
000	best-effort	low
001	best-effort	high
010	best-effort	low
011	best-effort	high
100	best-effort	low
101	best-effort	high
110	network-control	low
111	network-control	high

Related Documentation

- Default Behavior Aggregate Classification on page 724
- Sample Behavior Aggregate Classification on page 725
- *Example: Configuring Behavior Aggregate Classifiers*

Understanding Packet Loss Priorities

Packet loss priorities (PLPs) allow you to set the priority for dropping packets. You can use the PLP setting to identify packets that have experienced congestion. Typically, you mark packets exceeding some service level with a high loss priority—that is, a greater likelihood of being dropped. You set PLP by configuring a classifier or a policer. The PLP is used later in the work flow to select one of the drop profiles used by random early detection (RED).

You can configure the PLP bit as part of a congestion control strategy. The PLP bit can be configured on an interface or in a filter. A packet for which the PLP bit is set has an increased probability of being dropped during congestion.

Related Documentation

- Classification Overview on page 720
- Default Behavior Aggregate Classification on page 724

- Sample Behavior Aggregate Classification on page 725
- *Example: Configuring Behavior Aggregate Classifiers*

Default Behavior Aggregate Classification

Table 101 on page 724 shows the forwarding class (FC) and packet loss priority (PLP) that are assigned by default to each well-known Differentiated Services (DiffServ) code point (DSCP). Although several DSCPs map to the expedited-forwarding (ef) and assured-forwarding (af) classes, by default no resources are assigned to these forwarding classes. All af classes other than af1*x* are mapped to best-effort, because RFC 2597, *Assured Forwarding PHB Group*, prohibits a node from aggregating classes. Assignment to the best-effort FC implies that the node does not support that class. You can modify the default settings through configuration.

Table 101: Default Behavior Aggregate Classification

DSCP and DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
ef	expedited-forwarding	low
af11	assured-forwarding	low
af12	assured-forwarding	high
af13	assured-forwarding	high
af21	best-effort	low
af22	best-effort	low
af23	best-effort	low
af31	best-effort	low
af32	best-effort	low
af33	best-effort	low
af41	best-effort	low
af42	best-effort	low
af43	best-effort	low
be	best-effort	low
cs1	best-effort	low
cs2	best-effort	low

Table 101: Default Behavior Aggregate Classification (continued)

DSCP and DSCP IPv6 Alias	Forwarding Class	Packet Loss Priority
cs3	best-effort	low
cs4	best-effort	low
cs5	best-effort	low
nc1/cs6	network-control	low
nc2/cs7	network-control	low
other	best-effort	low

Related Documentation

- Classification Overview on page 720
- Sample Behavior Aggregate Classification on page 725
- *Example: Configuring Behavior Aggregate Classifiers*
- Understanding Packet Loss Priorities on page 723

Sample Behavior Aggregate Classification

Table 102 on page 725 shows the device forwarding classes (FCs) associated with each well-known Differentiated Services (DiffServ) code point (DSCP) and the resources assigned to the output queues for a sample DiffServ CoS implementation. This example assigns expedited forwarding to queue 1 and a subset of the assured FCs (af*x*) to queue 2, and distributes resources among all four forwarding classes. Other DiffServ-based implementations are possible.

Table 102: Sample Behavior Aggregate Classification Forwarding Classes and Queues

DSCP and DSCP IPv6 Alias	DSCP and DSCP IPv6 Bits	Forwarding Class	Packet Loss Priority	Queue
ef	101110	expedited-forwarding	low	1
af11	001010	assured-forwarding	low	2
af12	001100	assured-forwarding	high	2
af13	001110	assured-forwarding	high	2
af21	010010	best-effort	low	0
af22	010100	best-effort	low	0
af23	010110	best-effort	low	0

Table 102: Sample Behavior Aggregate Classification Forwarding Classes and Queues (continued)

DSCP and DSCP IPv6 Alias	DSCP and DSCP IPv6 Bits	Forwarding Class	Packet Loss Priority	Queue
af31	011010	best-effort	low	0
af32	011100	best-effort	low	0
af33	011110	best-effort	low	0
af41	100010	best-effort	low	0
af42	100100	best-effort	low	0
af43	100110	best-effort	low	0
be	000000	best-effort	low	0
cs1	0010000	best-effort	low	0
cs2	010000	best-effort	low	0
cs3	011000	best-effort	low	0
cs4	100000	best-effort	low	0
cs5	101000	best-effort	low	0
nc1/cs6	110000=	network-control	low	3
nc2/cs7	111000=	network-control	low	3
other	—	best-effort	low	0

Related Documentation

- Classification Overview on page 720
- Default Behavior Aggregate Classification on page 724
- *Example: Configuring Behavior Aggregate Classifiers*
- Understanding Packet Loss Priorities on page 723

Example: Configuring Behavior Aggregate Classifiers on a Security Device

This example shows how to configure behavior aggregate classifiers for a device to determine forwarding treatment of packets.

- Requirements on page 727
- Overview on page 727

Requirements

Before you begin, determine the forwarding class and PLP that are assigned by default to each well-known DSCP that you want to configure for the behavior aggregate classifier. See "Default Behavior Aggregate Classification" on page 724.

Overview

You configure behavior aggregate classifiers to classify packets that contain valid DSCPs to appropriate queues. Once configured, you must apply the behavior aggregate classifier to the correct interfaces. You can override the default IP precedence classifier by defining a classifier and applying it to a logical interface. To define new classifiers for all code point types, include the **classifiers** statement at the **[edit class-of-service]** hierarchy level.

In this example, you set the DSCP behavior aggregate classifier to ba-classifier as the default DSCP map. You set a best-effort forwarding class as be-class, an expedited forwarding class as ef-class, an assured forwarding class as af-class, and a network control forwarding class as nc-class. Finally, you apply the behavior aggregate classifier to an IRB interface.

Table 103 on page 727 shows how the behavior aggregate classifier assigns loss priorities, to incoming packets in the four forwarding classes.

Table 103: Sample ba-classifier Loss Priority Assignments

mf-classifier Forwarding Class	For CoS Traffic Type	ba-classifier Assignments
be-class	Best-effort traffic	High-priority code point: **000001**
ef-class	Expedited forwarding traffic	High-priority code point: **101111**
af-class	Assured forwarding traffic	High-priority code point: **001100**
nc-class	Network control traffic	High-priority code point: **110001**

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from the configuration mode.

```
set class-of-service classifiers dscp ba-classifier import default
set class-of-service classifiers dscp ba-classifier forwarding-class be-class loss-priority
    high code-points 000001
set class-of-service classifiers dscp ba-classifier forwarding-class ef-class loss-priority
    high code-points 101111
```

```
set class-of-service classifiers dscp ba-classifier forwarding-class af-class loss-priority
  high code-points 001100
set class-of-service classifiers dscp ba-classifier forwarding-class nc-class loss-priority
  high code-points 110001
set class-of-service interfaces irb unit 0 classifiers dscp ba-classifier
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the CLI User Guide.

To configure behavior aggregate classifiers for a device:

1. Configure the class of service.

   ```
   [edit]
   user@host# edit class-of-service
   ```

2. Configure behavior aggregate classifiers for DiffServ CoS.

   ```
   [edit class-of-service]
   user@host# edit classifiers dscp ba-classifier
   user@host# set import default
   ```

3. Configure a best-effort forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class be-class loss-priority high code-points 000001
   ```

4. Configure an expedited forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class ef-class loss-priority high code-points 101111
   ```

5. Configure an assured forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class af-class loss-priority high code-points 001100
   ```

6. Configure a network control forwarding class classifier.

   ```
   [edit class-of-service classifiers dscp ba-classifier]
   user@host# set forwarding-class nc-class loss-priority high code-points 110001
   ```

7. Apply the behavior aggregate classifier to an IRB interface.

   ```
   [edit]
   user@host# set class-of-service interfaces irb unit 0 classifiers dscp ba-classifier
   ```

NOTE: You can use interface wildcards for interface-name and logical-unit-number.

Results From configuration mode, confirm your configuration by entering the **show class-of-service** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show class-of-service
classifiers {
  dscp ba-classifier {
    import default;
    forwarding-class be-class {
      loss-priority high code-points 000001;
    }
    forwarding-class ef-class {
      loss-priority high code-points 101111;
    }
    forwarding-class af-class {
      loss-priority high code-points 001100;
    }
    forwarding-class nc-class {
      loss-priority high code-points 110001;
    }
  }
  forwarding-classes {
    class BE-data queue-num 0;
    class Premium-data queue-num 1;
    class Voice queue-num 2;
    class NC queue-num 3;
  }
  interfaces {
    irb {
      unit 0 {
        classifiers {
          dscp ba-classifier;
        }
      }
    }
    irb {
      unit 0 {
        classifiers {
          dscp v4-ba-classifier;
        }
        irb {
          unit 0 {
            classifiers {
              dscp v4-ba-classifier;
            }
            irb {
              unit 0 {
                classifiers {
                  dscp v4-ba-classifier;
                }
              }
            }
          }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying the Code-Point Aliases

Purpose Make sure that the code-point aliases are configured as expected.

Action Run the **show class-of-service code-point-aliases dscp** command.

```
user@host> show class-of-service code-point-aliases dscp

Code point type: dscp
  Alias               Bit pattern
  af11                001010
  af12                001100
  af13                001110
  af21                010010
  af22                010100
  af23                010110
  af31                011010
  af32                011100
  af33                011110
  af41                100010
  af42                100100
  af43                100110
  be                  000000
  be1                 000001
  cs1                 001000
  cs2                 010000
  cs3                 011000
  cs4                 100000
  cs5                 101000
  cs6                 110000
  cs7                 111000
  ef                  101110
  ef1                 101111
  nc1                 110000
  nc2                 111000
```

Meaning The code-point aliases are configured as expected. Note that the custom aliases that you configure are added to the default code-point aliases.

Verifying the DSCP Classifier

Purpose Make sure that the DSCP classifier is configured as expected.

Action Run the **show class-of-service classifiers name v4-ba-classifier** command.

```
user@host> show class-of-service classifiers name v4-ba-classifier

Classifier: v4-ba-classifier, Code point type: dscp, Index: 10755
  Code point         Forwarding class                  Loss priority
  000000             BE-data                           high
  000001             BE-data                           low
  000010             BE-data                           low
  000011             BE-data                           low
  000100             BE-data                           low
  000101             BE-data                           low
  000110             BE-data                           low
  000111             BE-data                           low
  001000             BE-data                           low
  001001             BE-data                           low
  001010             Voice                             low
  001011             BE-data                           low
  001100             Voice                             high
  001101             BE-data                           low
  001110             Voice                             high
  001111             BE-data                           low
  010000             BE-data                           low
  010001             BE-data                           low
  010010             BE-data                           low
  010011             BE-data                           low
  010100             BE-data                           low
  010101             BE-data                           low
  010110             BE-data                           low
  010111             BE-data                           low
  011000             BE-data                           low
  011001             BE-data                           low
  011010             BE-data                           low
  011011             BE-data                           low
  011100             BE-data                           low
  011101             BE-data                           low
  011110             BE-data                           low
  011111             BE-data                           low
  100000             BE-data                           low
  100001             BE-data                           low
  100010             BE-data                           low
  100011             BE-data                           low
  100100             BE-data                           low
  100101             BE-data                           low
  100110             BE-data                           low
  100111             BE-data                           low
  101000             BE-data                           low
  101001             BE-data                           low
  101010             BE-data                           low
  101011             BE-data                           low
  101100             BE-data                           low
  101101             BE-data                           low
  101110             Premium-data                      high
  101111             Premium-data                      low
```

```
  110000              NC                                low
  110001              BE-data                           low
  110010              BE-data                           low
  110011              BE-data                           low
  110100              BE-data                           low
  110101              BE-data                           low
  110110              BE-data                           low
  110111              BE-data                           low
  111000              NC                                low
  111001              BE-data                           low
  111010              BE-data                           low
  111011              BE-data                           low
  111100              BE-data                           low
  111101              BE-data                           low
  111110              BE-data                           low
  111111              BE-data                           low
```

Meaning Notice that the default classifier is incorporated into the customer classifier. If you were to remove the **import default** statement from the custom classifier, the custom classifier would look like this:

```
user@host> show class-of-service classifier name v4-ba-classifier
Classifier: v4-ba-classifier, Code point type: dscp, Index: 10755
  Code point         Forwarding class                  Loss priority
  000000             BE-data                           high
  000001             BE-data                           low
  101110             Premium-data                      high
  101111             Premium-data                      low
```

Verifying the Forwarding Classes and Output Queues

Purpose Make sure that the forwarding classes are configured as expected.

Action Run the **show class-of-service forwarding-class** command.

```
user@host> show class-of-service forwarding-class

Forwarding class                          ID      Queue  Restricted queue  Fabric
priority  Policing priority   SPU priority
  BE-data                                 0       0          0             low
            normal            low
  Premium-data                            1       1          1             low
            normal            low
  Voice                                   2       2          2             low
            normal            low
  NC                                      3       3          3             low
            normal            low
```

Meaning The forwarding classes are configured as expected.

Verifying That the Classifier Is Applied to the Interfaces

Purpose Make sure that the classifier is applied to the correct interfaces.

Action Run the **show class-of-service interface** command.

```
user@host> show class-of-service interface irb

Physical interface: irb, Index: 144
Queues supported: 8, Queues in use: 4
  Scheduler map: <default>, Index: 2
  Congestion-notification: Disabled

  Logical interface: irb, Index: 333
Object                 Name                 Type                 Index
Classifier             v4-ba-classifier     dscp                 10755
```

Meaning The interfaces are configured as expected.

Related Documentation

- *Interfaces Feature Guide for Security Devices*
- Classification Overview on page 720
- Sample Behavior Aggregate Classification on page 725
- Understanding Packet Loss Priorities on page 723

Example: Configuring and Applying a Firewall Filter for a Multifield Classifier

This example shows how to configure a firewall filter to classify traffic using a multifield classifier. The classifier detects packets of interest to CoS as they arrive on an interface.

Requirements

One common way to detect packets of CoS interest is by source or destination address. The destination address is used in this example, but many other matching criteria for packet detection are available to firewall filters.

Overview

In this example, you configure the firewall filter mf-classifier. You create and name the assured forwarding traffic class, set the match condition, and specify the destination address as 192.168.44.55. You create the forwarding class for assured forwarding DiffServ traffic as af-class and set the loss priority to low.

Then you create and name the expedited forwarding traffic class, set the match condition, for the expedited forwarding traffic class, and specify the destination address as 192.168.66.77. You then create the forwarding class for expedited forwarding DiffServ traffic as ef-class and set the policer to ef-policer. Then you create and name the network-control traffic class and set the match condition.

You then create and name the forwarding class for the network control traffic class as nc-class. You create and name the forwarding class for the best-effort traffic class as be-class. Finally, you apply the multifield classifier firewall filter as an input filter on each customer-facing or host-facing that needs the filter. In this example, the interface is ge-0/0/0.

Configuration

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set firewall filter mf-classifier interface-specific
set firewall filter mf-classifier term assured-forwarding from destination-address
  192.168.44.55
set firewall filter mf-classifier term assured-forwarding then forwarding-class af-class
set firewall filter mf-classifier term assured-forwarding then loss-priority low
set firewall filter mf-classifier term expedited-forwarding from destination-address
  192.168.66.77
set firewall filter mf-classifier term expedited-forwarding then forwarding-class ef-class
set firewall filter mf-classifier term expedited-forwarding then policer ef-policer
set firewall filter mf-classifier term network-control from precedence net-control
set firewall filter mf-classifier term network-control then forwarding-class nc-class
set firewall filter mf-classifier term best-effort then forwarding-class be-class
set interfaces irb unit 0 family inet filter input mf-classifier
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a firewall filter for a multifield classifier for a device:

1. Create and name the multifield classifier filter.

   ```
   [edit]
   user@host# edit firewall filter mf-classifier
   user@host# set interface-specific
   ```

2. Create and name the term for the assured forwarding traffic class.

   ```
   [edit firewall filter mf-classifier]
   user@host# edit term assured-forwarding
   ```

3. Specify the destination address for assured forwarding traffic.

   ```
   [edit firewall filter mf-classifier term assured-forwarding]
   ```

```
user@host# set from destination-address 192.168.44.55
```

4. Create the forwarding class and set the loss priority for the assured forwarding traffic class.

```
[edit firewall filter mf-classifier term assured-forwarding]
user@host# set then forwarding-class af-class
user@host# set then loss-priority low
```

5. Create and name the term for the expedited forwarding traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term expedited-forwarding
```

6. Specify the destination address for the expedited forwarding traffic.

```
[edit firewall filter mf-classifier term expedited-forwarding]
user@host# set from destination-address 192.168.66.77
```

7. Create the forwarding class and apply the policer for the expedited forwarding traffic class.

```
[edit firewall filter mf-classifier term expedited-forwarding]
user@host# set then forwarding-class ef-class
user@host# set then policer ef-policer
```

8. Create and name the term for the network control traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term network-control
```

9. Create the match condition for the network control traffic class.

```
[edit firewall filter mf-classifier term network-control]
user@host# set from precedence net-control
```

10. Create and name the forwarding class for the network control traffic class.

```
[edit firewall filter mf-classifier term network-control]
user@host# set then forwarding-class nc-class
```

11. Create and name the term for the best-effort traffic class.

```
[edit]
user@host# edit firewall filter mf-classifier
user@host# edit term best-effort
```

12. Create and name the forwarding class for the best-effort traffic class.

```
[edit firewall filter mf-classifier term best-effort]
user@host# set then forwarding-class be-class
```

NOTE: Because this is the last term in the filter, it has no match condition.

13. Apply the multifield classifier firewall filter as an input filter.

```
[edit]
user@host# set interfaces irb unit 0 family inet filter input mf-classifier
```

Results From configuration mode, confirm your configuration by entering the **show firewall filter mf-classifier** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show firewall filter mf-classifier
  interface-specific;
  term assured-forwarding {
  from {
    destination-address {
    192.168.44.55;
  }
  }
  then {
    loss-priority low;
    forwarding-class af-class;
  }
  }
  term expedited-forwarding {
    from {
    destination-address {
    192.168.66.77;
  }
  }
  then {
    policer ef-policer;
    forwarding-class ef-class;
  }
  }
  term network-control {
    from {
  precedence net-control;
  }
  then forwarding-class nc-class;
    }
    term best-effort {
  then forwarding-class be-class;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying a Firewall Filter for a Multifield Classifier Configuration

Purpose Verify that a firewall filter for a multifield classifier is configured properly on a device.

Action From configuration mode, enter the **show firewall filter mf-classifier** command.

Related Documentation
- Understanding Junos OS CoS Components for SRX Series Devices on page 718

Single-Rate Three-Color Policer Overview

A single-rate three-color policer defines a bandwidth limit and a maximum burst size for guaranteed traffic and a second burst size for peak traffic. A single-rate three-color policer is most useful when a service is structured according to packet length and not peak arrival rate.

Single-rate three-color policing meters a traffic stream based on the following configured traffic criteria:

- Committed information rate (CIR)—Bandwidth limit for guaranteed traffic.
- Committed burst size (CBS)—Maximum packet size permitted for bursts of data that exceed the CIR.
- Excess burst size (EBS)—Maximum packet size permitted for peak traffic.

Single-rate tricolor marking (single-rate TCM) classifies traffic as belonging to one of three color categories and performs congestion-control actions on the packets based on the color marking:

- Green—Traffic that conforms to *either* the bandwidth limit *or* the burst size for guaranteed traffic (CIR or CBS). For a green traffic flow, single-rate marks the packets with an implicit loss priority of **low** and transmits the packets.
- Yellow—Traffic that exceeds *both* the bandwidth limit *and* the burst size for guaranteed traffic (CIR and CBS) but not the burst size for peak traffic (EBS). For a yellow traffic flow, single-rate marks the packets with an implicit loss priority of **medium-high** and transmits the packets.
- Red—Traffic that exceeds the burst size for peak traffic (EBS), single-rate marks packets with an implicit loss priority of **high** and, optionally, discards the packets.

If congestion occurs downstream, the packets with higher loss priority are more likely to be discarded.

Related Documentation
- Example: Configuring a Single-Rate Three-Color Policer on a Security Device on page 738

Example: Configuring a Single-Rate Three-Color Policer on a Security Device

This example shows how to configure a single-rate three-color policer.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

A single-rate three-color policer meters a traffic flow against a bandwidth limit and burst-size limit for guaranteed traffic, plus a second burst-size limit for excess traffic. Traffic that conforms to the limits for guaranteed traffic is categorized as green, and nonconforming traffic falls into one of two categories:

- Nonconforming traffic that does not exceed the burst size for excess traffic is categorized as yellow.
- Nonconforming traffic that exceeds the burst size for excess traffic is categorized as red.

Each category is associated with an action. For green traffic, packets are implicitly set with a loss-priority value of **low** and then transmitted. For yellow traffic, packets are implicitly set with a loss-priority value of **medium-high** and then transmitted. For red traffic, packets are implicitly set with a loss-priority value of **high** and then transmitted. If the policer configuration includes the optional **action** statement (**action loss-priority high then discard**), then packets in a red flow are discarded instead.

You can apply a three-color policer to Layer 3 traffic as a firewall filter policer only. You reference the policer from a stateless firewall filter term, and then you apply the filter to the input or output of a logical interface at the protocol level.

Topology

In this example, you apply a color-aware, single-rate three-color policer to the input IPv4 traffic at IRB interface. The IPv4 firewall filter term that references the policer does not apply any packet-filtering. The filter is used only to apply the three-color policer to the interface.

You configure the policer to rate-limit traffic to a bandwidth limit of 40 Mbps and a burst-size limit of 100 KB for green traffic, but also allow an excess burst-size limit of 200 KB for yellow traffic. Only nonconforming traffic that exceeds the peak burst-size limit is categorized as red. In this example, you configure the three-color policer action **loss-priority high then discard**, which overrides the implicit marking of red traffic to a **high** loss priority.

Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*.

To configure this example, perform the following tasks:

- Configuring a Single-Rate Three-Color Policer on page 739
- Applying the Filter to the Logical Interface on page 740

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set firewall three-color-policer srTCM1-ca single-rate color-aware
set firewall three-color-policer srTCM1-ca single-rate committed-information-rate 40m
set firewall three-color-policer srTCM1-ca single-rate committed-burst-size 100k
set firewall three-color-policer srTCM1-ca single-rate excess-burst-size 200k
set firewall three-color-policer srTCM1-ca action loss-priority high then discard
set firewall family inet filter filter-srtcm1ca-all term 1 then three-color-policer single-rate
    srTCM1-ca
set class-of-service interfaces irb unit 0 forwarding-class af
set interfaces irb unit 0 family inet address 10.20.130.1/24
set interfaces irb unit 0 family inet filter input filter-srtcm1ca-all
```

Configuring a Single-Rate Three-Color Policer

Step-by-Step Procedure

To configure a single-rate three-color policer:

1. Enable configuration of a three-color policer.

   ```
   [edit]
   user@host# edit firewall three-color-policer srTCM1-ca
   ```

2. Configure the color mode of the single-rate three-color policer.

   ```
   [edit firewall three-color-policer srTCM1-ca]
   user@host# set single-rate color-aware
   ```

3. Configure the single-rate guaranteed traffic limits.

   ```
   [edit firewall three-color-policer srTCM1-ca]
   user@host# set single-rate committed-information-rate 40m
   user@host# set single-rate committed-burst-size 100k
   ```

4. Configure the single-rate burst-size limit that is used to classify nonconforming traffic.

   ```
   [edit firewall three-color-policer srTCM1-ca]
   user@host# set single-rate excess-burst-size 200k
   ```

5. (Optional) Configure the action for nonconforming traffic.

```
[edit firewall three-color-policer srTCM1-ca]
user@host# set action loss-priority high then discard
```

For three-color policers, the only configurable action is to discard packets in a red traffic flow. In this example, packets in a red traffic flow have been implicitly marked with a **high** packet loss priority (PLP) level because the traffic flow exceeded the rate-limiting defined by the single rate-limit (specified by the **committed-information-rate 40m** statement) and the larger burst-size limit (specified by the **excess-burst-size 200k** statement). Because the optional **action** statement is included, this example takes the more severe action of discarding packets in a red traffic flow.

Results Confirm the configuration of the hierarchical policer by entering the **show firewall** configuration command. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
three-color-policer srTCM1-ca {
    action {
        loss-priority high then discard;
    }
    single-rate {
        color-aware;
        committed-information-rate 40m;
        committed-burst-size 100k;
        excess-burst-size 200k;
    }
}
```

Applying the Filter to the Logical Interface

Step-by-Step Procedure To apply the filter to the IRB interface:

1. Enable configuration of the IRB interface.

```
[edit]
user@host# edit interfaces irb unit 0 family inet
```

2. Configure an IP address.

```
[edit interfaces irb unit 0 family inet]
user@host# set address 10.20.130.1/24
```

3. Reference the filter as an input filter.

```
[edit interfaces irb unit 0 family inet]
user@host# set filter input filter-srtcm1ca-all
```

Results Confirm the configuration of the interface by entering the **show class-of-service** and **show interfaces** configuration mode commands. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show class-of-service
interfaces {
  irb {
    unit 0 {
      forwarding-class af;
    }
  }
}
[edit]
user@host# show interfaces
irb {
  unit 0 {
    family inet {
      filter {
        input filter-srtcm1ca-all;
        }
      address 10.20.130.1/24;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Displaying the Firewall Filters Applied to the IRB Interface

Purpose Verify that the firewall filter is applied to the IRB interface.

Action Use the **show interfaces** operational mode command for the logical interface **ge-2/0/5.0**, and specify **detail** mode. The **Protocol inet** section of the command output displays information for the IRB interface.

```
user@host> show interfaces irb detail
  Physical interface irb (Index 105) (SNMP ifIndex 556) (Generation 170)
    Flags: Device-Down SNMP-Traps 0x4004000 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 inet, MTU: 1500, Generation: 242, Route table: 0
  Flags: Sendbcast-pkt-to-re
  Input Filters: filter-srtcm1ca-all
  Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
    Destination: 10.20.130/24, Local: 10.20.130.1, Broadcast: 10.20.130.255,

    Generation: 171
Protocol multiservice, MTU: Unlimited, Generation: 243, Route table: 0
  Policer: Input: __default_arp_policer__
```

Meaning The firewall filter is applied to the IRB interface as expected.

Related Documentation

- Single-Rate Three-Color Policer Overview on page 737

Rewrite Rules Overview

A rewrite rule modifies the appropriate class-of-service (CoS) bits in an outgoing packet. Modification of CoS bits allows the next downstream device to classify the packet into the appropriate service group. Rewriting or marking outbound packets is useful when the device is at the border of a network and must alter the CoS values to meet the policies of the targeted peer. A rewrite rule examines the forwarding class and loss priority of a packet and sets its bits to a corresponding value specified in the rule.

Typically, a device rewrites CoS values in outgoing packets on the outbound interfaces of an edge device, to meet the policies of the targeted peer. After reading the current forwarding class and loss priority information associated with the packet, the transmitting device locates the chosen CoS value from a table, and writes this CoS value into the packet header.

NOTE:

- You can configure up to 32 IEEE 802.1p rewrite rules on each SRX5K-MPC on the SRX5600 and SRX5800 devices.
- Starting in Junos OS Release 18.2R1, you can configure 802.1p rewrite on logical VDSL interface, that is, pt interface).

Release History Table

Release	Description
18.2R1	Starting in Junos OS Release 18.2R1, you can configure 802.1p rewrite on logical VDSL interface, that is, **pt** interface).

Rewriting Frame Relay Headers

Assigning the Default Frame Relay Rewrite Rule to an Interface

For Juniper Networks device interfaces with Frame Relay encapsulation, you can rewrite the discard eligibility (DE) bit based on the loss priority of Frame Relay traffic. For each outgoing frame with the loss priority set to low, medium-low, medium-high, or high, you can set the DE bit CoS value to **0** or **1**. You can combine a Frame Relay rewrite rule with other rewrite rules on the same interface. For example, you can rewrite both the DE bit and MPLS EXP bit.

The default Frame Relay rewrite rule contains the following settings:

```
loss-priority low code-point 0;
loss-priority medium-low code-point 0;
loss-priority medium-high code-point 1;
loss-priority high code-point 1;
```

This default rule sets the DE CoS value to **0** for each outgoing frame with the loss priority set to low or medium-low. This default rule sets the DE CoS value to **1** for each outgoing frame with the loss priority set to medium-high or high.

To assign the default rule to an interface, include the **frame-relay-de default** statement at the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number* unit rewrite-rules]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
frame-relay-de default;
```

Defining a Custom Frame Relay Rewrite Rule

To define a custom Frame Relay rewrite rule, include the following statements at the **[edit class-of-service]** hierarchy level:

```
[edit class-of-service]
rewrite-rules {
    frame-relay-de rewrite-name {
        import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority level code-point (0 | 1);
        }
    }
}
```

A custom rewrite rule sets the DE bit to the **0** or **1** CoS value based on the assigned loss priority of low, medium-low, medium-high, or high for each outgoing frame.

The rule does not take effect until you apply it to a logical interface. To apply the rule to a logical interface, include the **frame-relay-de** ***map-name*** statement at the **[edit class-of-service interfaces interface** ***interface-name*** **unit** ***logical-unit-number*** **rewrite-rules]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number rewrite-rules]
frame-relay-de map-name;
```

Related Documentation

- Rewrite Rules Overview on page 742
- Example: Configuring and Applying Rewrite Rules on a Security Device on page 744

Example: Configuring and Applying Rewrite Rules on a Security Device

This example shows how to configure and apply rewrite rules for a device.

Requirements

Before you begin, create and configure the forwarding classes.

Overview

You can configure rewrite rules to replace CoS values on packets received from the customer or host with the values expected by other devices. You do not have to configure rewrite rules if the received packets already contain valid CoS values. Rewrite rules apply the forwarding class information and packet loss priority used internally by the device to establish the CoS value on outbound packets. After you configure rewrite rules, you must apply them to the correct interfaces.

In this example, you configure the rewrite rule for DiffServ CoS as rewrite-dscps. You specify the best-effort forwarding class as be-class, expedited forwarding class as ef-class, an assured forwarding class as af-class, and a network control class as nc-class. Finally, you apply the rewrite rule to an IRB interface.

NOTE: You can apply one rewrite rule to each logical interface.

Table 104 on page 745 shows how the rewrite rules replace the DSCPs on packets in the four forwarding classes.

Table 104: Sample rewrite-dscps Rewrite Rules to Replace DSCPs

mf-classifier Forwarding Class	For CoS Traffic Type	rewrite-dscps Rewrite Rules
be-class	Best-effort traffic—Provides no special CoS handling of packets. Typically, RED drop profile is aggressive and no loss priority is defined.	Low-priority code point: 000000 High-priority code point: 000001
ef-class	Expedited forwarding traffic—Provides low loss, low delay, low jitter, assured bandwidth, and end-to-end service. Packets can be forwarded out of sequence or dropped.	Low-priority code point: 101110 High-priority code point: 101111
af-class	Assured forwarding traffic—Provides high assurance for packets within the specified service profile. Excess packets are dropped.	Low-priority code point: 001010 High-priority code point: 001100
nc-class	Network control traffic—Packets can be delayed, but not dropped.	Low-priority code point: 110000 High-priority code point: 110001

NOTE: Forwarding classes can be configured in a DSCP rewriter and also as an action of an IDP policy to rewrite DSCP code points. To ensure that the forwarding class is used as an action in an IDP policy, it is important that you do not configure an IDP policy and interface-based rewrite rules with the same forwarding class.

Configuration

- [xref target has no title]

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from the configuration mode.

```
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class
    loss-priority low code-point 000000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class be-class
    loss-priority high code-point 000001
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
    low code-point 101110
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class ef-class loss-priority
    high code-point 101111
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
    low code-point 001010
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class af-class loss-priority
    high code-point 001100
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
    low code-point 110000
set class-of-service rewrite-rules dscp rewrite-dscps forwarding-class nc-class loss-priority
    high code-point 110001
```

```
set class-of-service interfaces irb unit 0 rewrite-rules dscp rewrite-dscps
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure and apply rewrite rules for a device:

1. Configure rewrite rules for DiffServ CoS.

```
[edit]
user@host# edit class-of-service
user@host# edit rewrite-rules dscp rewrite-dscps
```

2. Configure best-effort forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class be-class loss-priority low code-point 000000
user@host# set forwarding-class be-class loss-priority high code-point 000001
```

3. Configure expedited forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class ef-class loss-priority low code-point 101110
user@host# set forwarding-class ef-class loss-priority high code-point 101111
```

4. Configure an assured forwarding class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class af-class loss-priority low code-point 001010
user@host# set forwarding-class af-class loss-priority high code-point 001100
```

5. Configure a network control class rewrite rules.

```
[edit class-of-service rewrite-rules dscp rewrite-dscps]
user@host# set forwarding-class nc-class loss-priority low code-point 110000
user@host# set forwarding-class nc-class loss-priority high code-point 110001
```

6. Apply rewrite rules to an IRB interface.

```
[edit class-of-service]
user@host# set interfaces irb unit 0 rewrite-rules dscp rewrite-dscps
```

Results

From configuration mode, confirm your configuration by entering the **show class-of-service** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show class-of-service
  interfaces {
  irb {
    unit 0 {
```

```
    rewrite-rules {
    dscp rewrite-dscps;
      }
    }
  }
  }
  rewrite-rules {
    dscp rewrite-dscps {
    forwarding-class be-class {
    loss-priority low code-point 000000;
      loss-priority high code-point 000001;
      }
      forwarding-class ef-class {
        loss-priority low code-point 101110;
      loss-priority high code-point 101111;
    }
    forwarding-class af-class {
      loss-priority low code-point 001010;
    loss-priority high code-point 001100;
  }
  forwarding-class nc-class {
    loss-priority low code-point 110000;
    loss-priority high code-point 110001;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Rewrite Rules Configuration

Purpose Verify that rewrite rules are configured properly.

Action From configuration mode, enter the **show class-of-service** command.

```
user@host> show class-of-service
Physical interface: irb, Index: 130
 Maximum usable queues: 8, Queues in use: 4
 Scheduler map: <default> , Index: 2
Congestion-notification: Disabled

Logical interface: irb.10, Index: 71
Object                 Name                    Type                    Index
Classifier             ipprec-compatibility    ip                         13
```

Meaning Rewrite rules are configured on IRB interface as expected.

Related Documentation

- Rewrite Rules Overview on page 742

Code-Point Aliases Overview

A code-point alias assigns a name to a pattern of code-point bits. You can use this name instead of the bit pattern when you configure other class-of-service (CoS) components, such as classifiers, drop-profile maps, and rewrite rules.

When you configure classes and define classifiers, you can refer to the markers by alias names. You can configure user-defined classifiers in terms of alias names. If the value of an alias changes, it alters the behavior of any classifier that references it.

The following types of code points are supported by Junos operating system (OS):

- DSCP—Defines aliases for DiffServ code point (DSCP) IPv4 values.

  You can refer to these aliases when you configure classes and define classifiers.
- DSCP-IPv6—Defines aliases for DSCP IPv6 values.

  You can refer to these aliases when you configure classes and define classifiers.
- EXP—Defines aliases for MPLS EXP bits.

  You can map MPLS EXP bits to the device forwarding classes.
- inet-precedence—Defines aliases for IPv4 precedence values.

  Precedence values are modified in the IPv4 type-of-service (ToS) field and mapped to values that correspond to levels of service.

Related Documentation

- Default CoS Values and Aliases on page 748
- Example: Defining Code-Point Aliases for Bits on a Security Device on page 751

Default CoS Values and Aliases

Table 105 on page 749 shows the default mapping between the standard aliases and the bit values.

Table 105: Standard CoS Aliases and Bit Values

CoS Value Type	Alias	Bit Value
MPLS EXP	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111

Table 105: Standard CoS Aliases and Bit Values (continued)

CoS Value Type	Alias	Bit Value
DSCP and DSCP IPv6	ef	101110
	af11	001010
	af12	001100
	af13	001110
	af21	010010
	af22	010100
	af23	010110
	af31	011010
	af32	011100
	af33	011110
	af41	100010
	af42	100100
	af43	100110
	be	000000
	cs1	001000
	cs2	010000
	cs3	011000
	cs4	100000
	cs5	101000
	nc1/cs6	110000
	nc2/cs7	111000

Table 105: Standard CoS Aliases and Bit Values (continued)

CoS Value Type	Alias	Bit Value
IEEE 802.1	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111
IP precedence	be	000
	be1	001
	ef	010
	ef1	011
	af11	100
	af12	101
	nc1/cs6	110
	nc2/cs7	111

Related Documentation

- Code-Point Aliases Overview on page 748
- Example: Defining Code-Point Aliases for Bits on a Security Device on page 751

Example: Defining Code-Point Aliases for Bits on a Security Device

This example shows how to define code-point aliases for bits on a device.

- Requirements on page 752
- Overview on page 752
- Configuration on page 752
- Verification on page 752

Requirements

Before you begin, determine which default mapping to use. See "Default CoS Values and Aliases" on page 748.

Overview

In this example, you configure class of service and specify names and values for the CoS code-point aliases that you want to configure. Finally, you specify CoS value using the appropriate formats.

Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To define code-point aliases for bits on a device:

1. Configure class of service.

```
[edit]
user@host# edit class-of-service
```

2. Specify CoS values.

```
[edit class-of-service]
user@host# set code-point-aliases dscp my1 110001
user@host# set code-point-aliases dscp my2 101110
user@host# set code-point-aliases dscp be 000001
user@host# set code-point-aliases dscp cs7 110000
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show class-of-service code-point-aliases dscp** command.

Related Documentation

- Code-Point Aliases Overview on page 748

Schedulers Overview

You use schedulers to define the properties of output queues. These properties include the amount of interface bandwidth assigned to the queue, the size of the memory buffer allocated for storing packets, the priority of the queue, and the random early detection (RED) drop profiles associated with the queue.

You associate the schedulers with forwarding classes by means of scheduler maps. You can then associate each scheduler map with an interface, thereby configuring the hardware queues, packet schedulers, and RED processes that operate according to this mapping.

An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.

You can configure per-unit scheduling (also called logical interface scheduling) to allow multiple output queues on a logical interface and to associate an output scheduler with each queue.

NOTE: For Juniper Network devices, when configuring the *protocol parameter* in the **drop-profile-map** statement, TCP and non-TCP values are not supported; only the value *any* is supported.

This topic contains the following sections:

Transmit Rate

The transmission rate determines the traffic transmission bandwidth for each forwarding class you configure. The rate is specified in bits per second (bps). Each queue is allocated some portion of the bandwidth of the outgoing interface.

This bandwidth amount can be a fixed value, such as 1 megabit per second (Mbps), a percentage of the total available bandwidth, or the rest of the available bandwidth. You can limit the transmission bandwidth to the exact value you configure, or allow it to exceed the configured rate if additional bandwidth is available from other queues (SRX5400, SRX5600, and SRX5800 devices do not support an exact value transmit rate). This property helps ensure that each queue receives the amount of bandwidth appropriate to its level of service.

The minimum transmit rate supported on high-speed interfaces is one-ten thousandth of the speed of that interface. For example, on a Gigabit Ethernet interface with a speed of 1000 Mbps, the minimum transmit rate is 100 Kbps (1,000 Mbps x 1/10,000). You can configure transmit rates in the range 3200 bps through 160,000,000,000 bps. When the configured rate is less than the minimum transmit rate, the minimum transmit rate is used instead.

NOTE: Interfaces with slower interface speeds, like T1, E1, or channelized T1/E1/ISDN PRI, cannot support minimum transmit rates because the minimum transmit rate supported on a device is 3,200 bps.

Transmit rate assigns the weighted round-robin (WRR) priority values within a given priority level and not between priorities.

The transmit rate defines the transmission rate of a scheduler. The transmit rate determines the traffic bandwidth from each forwarding class you configure.

By default, queues 0 through 7 have the following percentage of transmission capacity:

- Queue 0—95 percent
- Queue 1—0 percent
- Queue 2—0 percent
- Queue 3—0 percent
- Queue 4—0 percent
- Queue 6—0 percent
- Queue 7—5 percent

To define a transmit rate, select the appropriate option:

- To specify a transmit rate, select **rate** and type an integer from 3200 to 160,000,000,000 bits per second.
- To enforce an exact transmit rate, select **rate**.
- To specify the remaining transmission capacity, select **remainder**.
- To specify a percentage of transmission capacity, select **percent** and type an integer from 1 through 100.

Optionally, you can specify the percentage of the remainder to be used for allocating the transmit rate of the scheduler on a prorated basis. If there are still points left even after allocating the remainder percentage with the transmit rate and there are no queues, then the points are allocated point by point to each queue in a round-robin method. If the remainder percentage is not specified, the remainder value will be shared equally.

Delay Buffer Size

You can configure the delay buffer size to control congestion at the output stage. A delay buffer provides packet buffer space to absorb burst traffic up to a specified duration of delay. When the buffer is full, all packets are dropped.

On Juniper Networks devices, you can configure larger delay buffers on channelized T1/E1 interfaces. Larger delay buffers help these slower interfaces to avoid congestion and packet dropping when they receive large bursts of traffic.

By default, SRX300, SRX320, SRX340, SRX345, and SRX550M device interfaces support a delay buffer time of 100,000 microseconds. (Platform support depends on the Junos OS release in your implementation.)

To define a delay buffer size for a scheduler, select the appropriate option:

- To enforce exact buffer size, select **Exact**.
- To specify a buffer size as a temporal value (microseconds), select **Temporal**.
- To specify buffer size as a percentage of the total buffer, select **Percent** and type an integer from 1 through 100.
- To specify buffer size as the remaining available buffer, select **Remainder**.

Optionally, you can specify the percentage of the remainder to be used for allocating the buffer size of the scheduler on a prorated basis.

By default, sizes of the delay buffer queues 0 through 7 have the following percentage of the total available buffer space:

- Queue 0—95 percent
- Queue 1—0 percent
- Queue 2—0 percent
- Queue 3—0 percent
- Queue 4—0 percent
- Queue 5—0 percent
- Queue 6—0 percent
- Queue 7—5 percent

NOTE: A large buffer size value correlates with a greater possibility of packet delays. This might not be practical for sensitive traffic such as voice or video.

NOTE: For a Juniper Networks device, if the buffer size percentage is set to zero for T1 interfaces, traffic does not pass.

Scheduling Priority

Scheduling priority determines the order in which an output interface transmits traffic from the queues, thus ensuring that queues containing important traffic are provided better access to the outgoing interface.

The queues for an interface are divided into sets based on their priority. Each set contains queues of the same priority. The device examines the sets in descending order of priority. If at least one queue in a set has a packet to transmit, the device selects that set. If multiple queues in the set have packets to transmit, the device selects a queue from the

set according to the weighted round-robin (WRR) algorithm that operates within the set.

The packets in a queue are transmitted based on the configured scheduling priority, the transmit rate, and the available bandwidth.

The scheduling priority of the scheduler determines the order in which an output interface transmits traffic from the queues. You can set scheduling priority at different levels in an order of increasing priority from low to high. A high-priority queue with a high transmission rate might lock out lower-priority traffic.

To specify a scheduling priority, select one of the following levels:

- **high**—Packets in this queue have high priority.
- **low**—Packets in this queue are transmitted last.
- **medium—low**—Packets in this queue have medium-low priority.
- **medium—high**—Packets in this queue have medium-high priority.
- **strict—high**—Packets in this queue are transmitted first.

Shaping Rate

Shaping rates control the maximum rate of traffic transmitted on an interface. You can configure the shaping rate so that the interface transmits less traffic than it is physically capable of carrying.

You can configure shaping rates on logical interfaces. By default, output scheduling is not enabled on logical interfaces. Logical interface scheduling (also called per-unit scheduling) allows you to enable multiple output queues on a logical interface and associate an output scheduler and shaping rate with the queues.

By default, the logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. You can specify a peak bandwidth rate in bits per second (bps), either as a complete decimal number or as a decimal number followed by the abbreviation *k* (1000), *m* (1,000,000), or *g* (1,000,000,000). The range is from 1000 through 32,000,000,000 bps.

For low-speed interfaces, the queue-limit values might become lower than the interface MTU so that traffic with large packets can no longer pass through some of the queues. If you want larger-sized packets to flow through, set the buffer-size configuration in the scheduler to a larger value. For more accuracy, the 100-ms queue-limit values are calculated based on shaping rate and not on interface rates.

The shaping rate defines the minimum bandwidth allocated to a queue. The default shaping rate is 100 percent, which is the same as no shaping at all. To define a shaping rate, select the appropriate option:

- To specify shaping rate as an absolute number of bits per second, select **rate** and type an integer from 3200 to 160,000,000,000 bits per second.

- To specify shaping rate as a percentage, select **percent** and type an integer from 0 through 100.

Related Documentation

- *Default Scheduler Settings*
- Example: Configuring Class-of-Service Schedulers on a Security Device on page 757
- *Scheduler Buffer Size Overview*
- *Example: Configuring a Large Delay Buffer on a Channelized T1 Interface*
- *Example: Configuring and Applying Scheduler Maps*
- *Transmission Scheduling Overview*

Example: Configuring Class-of-Service Schedulers on a Security Device

This example shows how to configure CoS schedulers on a device.

Requirements

Before you begin, determine the buffer size allocation method to use. See *Scheduler Buffer Size Overview*.

Overview

An individual device interface has multiple queues assigned to store packets temporarily before transmission. To determine the order in which to service the queues, the device uses a round-robin scheduling method based on priority and the queue's weighted round-robin (WRR) credits. Junos OS schedulers allow you to define the priority, bandwidth, delay buffer size, rate control status, and RED drop profiles to be applied to a particular queue for packet transmission.

You configure schedulers to assign resources, priorities, and drop profiles to output queues. By default, only queues 0 and 3 have resources assigned.

NOTE: Juniper Network devices support hierarchical schedulers, including per-unit schedulers.

In this example, you configure a best-effort scheduler called be-scheduler. You set the priority as low and the buffer size to 40. You set the be-scheduler transmit-rate remainder percentage to 40. You configure an expedited forwarding scheduler called ef-scheduler and set the priority as high and the buffer size to 10. You set the ef-scheduler transmit-rate remainder percentage to 50.

Then you configure an assured forwarding scheduler called af-scheduler and set the priority as high and buffer size to 45. You set an assured forwarding scheduler transmit rate to 45. You then configure a drop profile map for assured forwarding as low and high priority. (DiffServ can have a RED drop profile associated with assured forwarding.)

Finally, you configure a network control scheduler called nc-scheduler and set the priority as low and buffer size to 5. You set a network control scheduler transmit rate to 5.

Table 106 on page 758 shows the schedulers created in this example.

Table 106: Sample Schedulers

Scheduler	For CoS Traffic Type	Assigned Priority	Allocated Portion of Queue Buffer	Allocated Portion of Remainder (Transmit Rate)
be-scheduler	Best-effort traffic	Low	40 percent	40 percent
ef-scheduler	Expedited forwarding traffic	High	10 percent	50 percent
af-scheduler	Assured forwarding traffic	High	45 percent	—
nc-scheduler	Network control traffic	Low	5 percent	—

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from the configuration mode.

```
set class-of-service schedulers be-scheduler priority low buffer-size percent 40
set class-of-service schedulers be-scheduler transmit-rate remainder 40
set class-of-service schedulers ef-scheduler priority high buffer-size percent 10
set class-of-service schedulers ef-scheduler transmit-rate remainder 50
set class-of-service schedulers af-scheduler priority high buffer-size percent 45
set class-of-service schedulers af-scheduler transmit-rate percent 45
set class-of-service schedulers af-scheduler drop-profile-map loss-priority low protocol
    any drop-profile af-normal
set class-of-service schedulers af-scheduler drop-profile-map loss-priority high protocol
    any drop-profile af-with-PLP
set class-of-service schedulers nc-scheduler priority low buffer-size percent 5
set class-of-service schedulers nc-scheduler transmit-rate percent 5
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure CoS schedulers:

1. Configure a best-effort scheduler.

```
[edit]
user@host# edit class-of-service schedulers be-scheduler
```

2. Specify a best-effort scheduler priority and buffer size.

```
[edit class-of-service schedulers be-scheduler]
user@host# set priority low
user@host# set buffer-size percent 40
```

3. Configure a remainder option for a best-effort scheduler transmit rate.

```
[edit class-of-service schedulers be-scheduler]
user@host# set transmit-rate remainder 40
```

4. Configure an expedited forwarding scheduler.

```
[edit]
user@host# edit class-of-service schedulers ef-scheduler
```

5. Specify an expedited forwarding scheduler priority and buffer size.

```
[edit class-of-service schedulers ef-scheduler]
user@host# set priority high
user@host# set buffer-size percent 10
```

6. Configure a remainder option for an expedited forwarding scheduler transmit rate.

```
[edit class-of-service schedulers ef-scheduler]
user@host# set transmit-rate remainder 50
```

7. Configure an assured forwarding scheduler.

```
[edit]
user@host# edit class-of-service schedulers af-scheduler
```

8. Specify an assured forwarding scheduler priority and buffer size.

```
[edit class-of-service schedulers af-scheduler]
user@host# set priority high
user@host# set buffer-size percent 45
```

9. Configure an assured forwarding scheduler transmit rate.

```
[edit class-of-service schedulers af-scheduler]
user@host# set transmit-rate percent 45
```

10. Configure a drop profile map for assured forwarding low and high priority.

```
[edit class-of-service schedulers af-scheduler]
user@host# set drop-profile-map loss-priority low protocol any drop-profile
  af-normal
user@host# set drop-profile-map loss-priority high protocol any drop-profile
  af-with-PLP
```

11. Configure a network control scheduler.

```
[edit]
user@host# edit class-of-service schedulers nc-scheduler
```

12. Specify a network control scheduler priority and buffer size.

```
[edit class-of-service schedulers nc-scheduler]
user@host# set priority low
user@host# set buffer-size percent 5
```

13. Configure a network control scheduler transmit rate.

```
[edit class-of-service schedulers nc-scheduler]
user@host# set transmit-rate percent 5
```

Results From configuration mode, confirm your configuration by entering the **show class-of-service** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show class-of-service
  schedulers {
  be-scheduler {
  transmit-rate remainder 40;
    buffer-size percent 40;
  priority low;
  }
  ef-scheduler {
    transmit-rate remainder 50;
    buffer-size percent 10;
  priority high;
  }
  af-scheduler {
    transmit-rate percent 45;
    buffer-size percent 45;
    priority high;
      drop-profile-map loss-priority low protocol any drop-profile af-normal;
    drop-profile-map loss-priority high protocol any drop-profile af-with-PLP;
    }
    nc-scheduler {
      transmit-rate percent 5;
      buffer-size percent 5;
    priority low;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Schedulers Configuration

Purpose Verify that the schedulers are configured properly.

Action From operational mode, enter the **show class-of-service** command.

Related Documentation

- Schedulers Overview on page 752
- *Default Scheduler Settings*
- *Example: Configuring a Large Delay Buffer on a Channelized T1 Interface*
- *Example: Configuring and Applying Scheduler Maps*
- *Transmission Scheduling Overview*

Virtual Channels Overview

You can configure virtual channels to limit traffic sent from a corporate headquarters to its branch offices. Virtual channels might be required when the headquarters site has an expected aggregate bandwidth higher than that of the individual branch offices. The headquarters router must limit the traffic sent to each branch office router to avoid oversubscribing their links. For instance, if branch 1 has a 1.5 Mbps link and the headquarters router attempts to send 6 Mbps to branch 1, all of the traffic in excess of 1.5-Mbps is dropped in the ISP network.

You configure virtual channels on a logical interface. Each virtual channel has a set of eight queues with a scheduler and an optional shaper. You can use an output firewall filter to direct traffic to a particular virtual channel. For example, a filter can direct all traffic with a destination address for branch office 1 to virtual channel 1, and all traffic with a destination address for branch office 2 to virtual channel 2.

Although a virtual channel group is assigned to a logical interface, a virtual channel is quite different from a logical interface. The only features supported on a virtual channel are queuing, packet scheduling, and accounting. Rewrite rules and routing protocols apply to the entire logical interface.

When you configure virtual channels on an interface, the virtual channel group uses the same scheduler and shaper you configure at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level. In this way, virtual channels are an extension of regular scheduling and shaping and are not independent entities.

Related Documentation

- Understanding Virtual Channels on page 762
- Example: Configuring Virtual Channels on a Security Device on page 763

Understanding Virtual Channels

You configure a virtual channel to set up queuing, packet scheduling, and accounting rules to be applied to one or more logical interfaces. You must apply then the virtual channel to a particular logical interface.

You also create a list of virtual channels that you can assign to a virtual channel group. To define a virtual channel group that you can assign to a logical interface, include the **virtual-channel-groups** statement at the **[edit class-of-service]** hierarchy level.

The *virtual-channel-group-name* can be any name that you want. The *virtual-channel-name* must be one of the names that you define at the **[edit class-of-service virtual-channels]** hierarchy level. You can include multiple virtual channel names in a group.

The scheduler map is required. The *map-name* must be one of the scheduler maps that you configure at the **[edit class-of-service scheduler-maps]** hierarchy level. For more information, see "Example: Configuring Class-of-Service Schedulers on a Security Device" on page 757.

The shaping rate is optional. If you configure the shaping rate as a percentage, when the virtual channel is applied to a logical interface, the shaping rate is set to the specified percentage of the interface bandwidth. If you configure a shaper on a virtual channel, the shaper limits the maximum bandwidth transmitted by that virtual channel. Virtual channels without a shaper can use the full logical interface bandwidth. If there are multiple unshaped virtual channels, they share the available logical interface bandwidth equally.

When you apply the virtual channel group to a logical interface, a set of eight queues is created for each of the virtual channels in the group. The **scheduler-map** statement applies a scheduler to these queues. If you include the **shaping-rate** statement, a shaper is applied to the entire virtual channel.

You must configure one of the virtual channels in the group to be the default channel. Therefore, the **default** statement is required in the configuration of one virtual channel per channel group. Any traffic not explicitly directed to a particular channel is transmitted by this default virtual channel.

For the corresponding physical interface, you must also include the **per-unit-scheduler** statement at the **[edit interfaces *interface-name*]** hierarchy level as follows:

```
[edit interfaces interface-name]
per-unit-scheduler;
```

The **per-unit-scheduler** statement enables one set of output queues for each logical interface configured under the physical interface.

When you apply a virtual channel group to a logical interface, the software creates a set of eight queues for each of the virtual channels in the group.

If you apply a virtual channel group to multiple logical interfaces, the software creates a set of eight queues on each logical interface. The virtual channel names listed in the group are used on all the logical interfaces. We recommend specifying the scheduler and

shaping rates in the virtual channel configuration in terms of percentages, rather than absolute rates. This allows you to apply the same virtual channel group to logical interfaces that have different bandwidths.

When you apply a virtual channel group to a logical interface, you cannot include the **scheduler-map** and **shaping-rate** statements at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level. In other words, you can configure a scheduler map and a shaping rate on a logical interface, or you can configure virtual channels on the logical interface, but not both.

If you configure multiple logical interfaces on a single physical interface, each logical interface is guaranteed an equal fraction of the physical interface bandwidth as follows:

```
logical-interface-bandwidth =
physical-interface-bandwidth / number-of-logical-interfaces
```

If one or more logical interfaces do not completely use their allocation, the other logical interfaces share the excess bandwidth equally.

If you configure multiple virtual channels on a logical interface, they are each guaranteed an equal fraction of the logical interface bandwidth as follows:

```
virtual-channel-bandwidth =
logical-interface-bandwidth / number-of-virtual-channels
```

If you configure a shaper on a virtual channel, the shaper limits the maximum bandwidth transmitted by that virtual channel. Virtual channels without a shaper can use the full logical interface bandwidth. If there are multiple unshaped virtual channels, they share the available logical interface bandwidth equally.

Related Documentation

- Virtual Channels Overview on page 761
- Example: Configuring Virtual Channels on a Security Device on page 763

Example: Configuring Virtual Channels on a Security Device

This example shows how to create virtual channels between a headquarters and its branch office.

Requirements

Before you begin, ensure that your headquarters and branch office have a network connection where the expected aggregate bandwidth is higher for your headquarters than for your branch office. The devices at your headquarters will then be set up to limit the traffic sent to the branch office to avoid oversubscribing the link.

Overview

In this example, you create the virtual channels as branch1–vc, branch2–vc, branch3–vc, and default-vc. You then define the virtual channel group as wan-vc-group to include the four virtual channels and assign the scheduler map as bestscheduler to each virtual channel. Three of the virtual channels are shaped to 1.5 Mbps. The fourth virtual channel is default-vc, and it is not shaped. Hence can use the full interface bandwidth.

Then you apply them in the firewall filter as choose-vc to the device's irb interface. The output filter on the interface sends all traffic with a destination address matching 192.168.10.0/24 to branch1-vc, and similar configurations are set for branch2-vc and branch3-vc. Traffic not matching any of the addresses goes to the default, unshaped virtual channel.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from the configuration mode.

```
set class-of-service virtual-channels branch1-vc
set class-of-service virtual-channels branch2-vc
set class-of-service virtual-channels branch3-vc
set class-of-service virtual-channels default-vc
set class-of-service virtual-channel-groups wan-vc-group branch1-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group branch2-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group branch3-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group default-vc scheduler-map
  bestscheduler
set class-of-service virtual-channel-groups wan-vc-group default-vc default
set class-of-service virtual-channel-groups wan-vc-group branch1-vc shaping-rate
  1500000
set class-of-service virtual-channel-groups wan-vc-group branch2-vc shaping-rate
  1500000
set class-of-service virtual-channel-groups wan-vc-group branch3-vc shaping-rate
  1500000
set class-of-service interfaces t3-1/0/0 unit 0 virtual-channel-group wan-vc-group
set firewall family inet filter choose-vc term branch1 from destination-address
  192.168.10.0/24
set firewall family inet filter choose-vc term branch1 then accept
set firewall family inet filter choose-vc term branch1 then virtual-channel branch1-vc
set firewall family inet filter choose-vc term branch1 then virtual-channel branch2-vc
set firewall family inet filter choose-vc term branch1 then virtual-channel branch3-vc
set interfaces irb unit 0 family inet filter output choose-vc
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure virtual channels:

1. Define the virtual channels and the default virtual channel.

```
[edit]
user@host# edit class-of-service
user@host# set virtual-channels branch1-vc
user@host# set virtual-channels branch2-vc
user@host# set virtual-channels branch3-vc
user@host# set virtual-channels default-vc
```

2. Define the virtual channel group and assign each virtual channel a scheduler map.

```
[edit class-of-service]
user@host# set virtual-channel-groups wan-vc-group branch1-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group branch2-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group branch3-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group default-vc scheduler-map
  bestscheduler
user@host# set virtual-channel-groups wan-vc-group default-vc default
```

3. Specify a shaping rate.

```
[edit class-of-service]
user@host# set virtual-channel-groups wan-vc-group branch1-vc shaping-rate 1.5m
user@host# set virtual-channel-groups wan-vc-group branch2-vc shaping-rate
  1.5m
user@host# set virtual-channel-groups wan-vc-group branch3-vc shaping-rate
  1.5m
```

4. Apply the virtual channel group to the irb interface.

```
[edit class-of-service]
user@host# set interfaces irb unit 0 virtual-channel-group wan-vc-group
```

5. Create the firewall filter to select the traffic.

```
[edit firewall]
user@host# set family inet filter choose-vc term branch1 from destination
  192.168.10.0/24
user@host# set family inet filter choose-vc term branch1 then accept
user@host# set family inet filter choose-vc term branch1 then virtual-channel
  branch1-vc
user@host# set family inet filter choose-vc term branch1 then virtual-channel
  branch2-vc
user@host# set family inet filter choose-vc term branch1 then virtual-channel
  branch3-vc
```

6. Apply the firewall filter to output traffic.

```
[edit interfaces]
user@host# set irb unit 0 family inet filter output choose-vc
```

Results From configuration mode, confirm your configuration by entering the **show class-of-service**, **show firewall**, and **show interfaces irb** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show class-of-service
virtual-channels {
  branch1-vc;
  branch2-vc;
  branch3-vc;
  default-vc;
}
virtual-channel-groups {
  wan-vc-group {
    branch1-vc {
    scheduler-map bestscheduler;
    shaping-rate 1500000;
  }
  branch2-vc {
    scheduler-map bestscheduler;
    shaping-rate 1500000;
  }
  branch3-vc {
    scheduler-map bestscheduler;
    shaping-rate 1500000;
  }
  default-vc {
    scheduler-map bestscheduler;
    default;
  }
}
}
interfaces {
  irb {
    unit 0 {
      virtual-channel-group wan-vc-group;
    }
  }
}
[edit]
user@host# show firewall
family inet {
  filter choose-vc {
    term branch1 {
      from {
        destination-address {
          192.168.10.0/24;
        }
      }
```

```
        then {
          virtual-channel branch3-vc;
          accept;
        }
      }
    }
  }
[edit]
user@host# show interfaces irb
unit 0 {
  family inet {
    filter {
      output choose-vc;
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying Virtual Channel Configuration

Purpose Verify that the virtual channels are properly configured.

Action From configuration mode, enter the **show class-of-service**, **show firewall**, and **show interfaces irb** commands.

Related Documentation

- Virtual Channels Overview on page 761
- Understanding Virtual Channels on page 762

CHAPTER 32

Configuring Ethernet Port VLANs in Switching Mode on Security Devices

Understanding VLANs

Each VLAN is a collection of network nodes that are grouped together to form separate broadcast domains. On an Ethernet network that is a single LAN, all traffic is forwarded to all nodes on the LAN. On VLANs, frames whose origin and destination are in the same VLAN are forwarded only within the local VLAN. Frames that are not destined for the local VLAN are the only ones 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 a VLAN and on the LAN as a whole.

On an Ethernet LAN, all network nodes must be physically connected to the same network. On VLANs, the physical location of the nodes is not important; therefore, you can group network devices in any way that makes sense for your organization, such as by department or business function, by types of network nodes, or even by physical location. Each VLAN is identified by a single IP subnetwork and by standardized IEEE 802.1Q encapsulation.

To identify which VLAN the traffic belongs to, all frames on an Ethernet VLAN are identified by a tag, as defined in the IEEE 802.1Q standard. These frames are tagged and are encapsulated with 802.1Q tags.

For a simple network that has only a single VLAN, all traffic has the same 802.1Q tag. 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 know to which VLAN a frame belongs. 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 VLAN configuration details, see Table 107 on page 770.

Table 107: VLAN Configuration Details

Field	Function	Action
General		
VLAN Name	Specifies a unique name for the VLAN.	Enter a name. NOTE: VLAN text field is disabled when VLAN tagging is not enabled.
VLAN ID/Range	Specifies the identifier or range for the VLAN.	Select one: • **VLAN ID**—Type a unique identification number from **1** through **4094**. If no value is specified, it defaults to 1. • **VLAN Range**—Type a number range to create VLANs with IDs corresponding to the range. For example, the range 2–3 will create two VLANs with the ID 2 and 3.
Description	Describes the VLAN.	Enter a brief description for the VLAN.
Input Filter	Specifies the VLAN firewall filter that is applied to incoming packets.	To apply an input firewall filter, select the firewall filter from the list.
Output Filter	Specifies the VLAN firewall filter that is applied to outgoing packets.	To apply an output firewall filter, select the firewall filter from the list.
Ports		
Ports	Specifies the ports to be associated with this VLAN for data traffic. You can also remove the port association.	Click one: • **Add**—Select the ports from the available list. • **Remove**—Select the port that you do not want associated with the VLAN.
IP Address		
Layer 3 Information	Specifies IP address options for the VLAN.	Select to enable the IP address options.
IP Address	Specifies the IP address of the VLAN.	Enter the IP address.
Subnet Mask	Specifies the range of logical addresses within the address space that is assigned to an organization.	Enter the address, for example, **203.0.113.0**. You can also specify the address prefix.
Input Filter	Specifies the VLAN interface firewall filter that is applied to incoming packets.	To apply an input firewall filter to an interface, select the firewall filter from the list.
Output Filter	Specifies the VLAN interface firewall filter that is applied to outgoing packets.	To apply an output firewall filter to an interface, select the firewall filter from the list.
ARP/MAC Details	Specifies the details for configuring the static IP address and MAC.	Click the **ARP/MAC Details** button. Enter the static IP address and MAC address in the window that is displayed.
VoIP		

Table 107: VLAN Configuration Details (continued)

Field	Function	Action
Ports	Specifies the ports to be associated with this VLAN for voice traffic. You can also remove the port association.	Click one: • **Add**—Select the ports from the available list. • **Remove**—Select the port that you do not want associated with the VLAN.

NOTE: On SRX100 devices, dynamic VLAN assignments and guest VLANs are not supported.

On SRX240, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX650 devices, the VLAN range from 1 to 4094 on inet interfaces and the VLAN range from 1 to 3967 on Ethernet switching interfaces. On Ethernet switching interfaces, the VLAN range from 3968 to 4094 falls under the reserved VLAN address range, and the user is not allowed to configure VLANs in this range.

Related Documentation

- *Example: Configuring VLANs on Security Devices (J-Web Procedure)*
- Ethernet Ports Switching Overview for Security Devices on page 706

Example: Configuring VLANs on Security Devices (CLI Procedure)

This example shows you how to configure a VLAN.

Requirements

Before you begin:

- Determine which interfaces to use and verify that they are in switching mode. See "Understanding VLANs" on page 769.
- Determine what ports to use on the device and how to segment your network. See "Ethernet Ports Switching Overview for Security Devices" on page 706.

Overview

In this example, you create a new VLAN and then configure its attributes. You can configure one or more VLANs to perform Layer 2 switching. The Layer 2 switching functions include integrated routing and bridging (IRB) for support for Layer 2 switching and Layer 3 IP

routing on the same interface. SRX Series devices can function as Layer 2 switches, each with multiple switching or broadcast domains that participate in the same Layer 2 network.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set vlans v10 vlan-id 10
set vlans v10 l3-interface irb.10
set interfaces irb unit 10 family inet address 10.1.1.10/24
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members 10
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a VLAN:

1. Create the VLAN by setting the unique VLAN name and configuring the VLAN ID.

   ```
   [edit vlans]
   user@host# set vlans v10 vlan-id 10
   ```

2. Bind a Layer 3 interface with the VLAN.

   ```
   [edit]
   user@host# set vlans v10 l3-interface irb.10
   ```

3. Create the subnet for the VLAN's broadcast domain.

   ```
   [edit]
   user@host# set interfaces irb unit 10 family inet address 10.1.1.10/24
   ```

4. 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@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
     10
   ```

Results

From configuration mode, confirm your configuration by entering the **show vlans** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show vlans
v10 {
   vlan-id 10;
   l3-interface irb.10;
```

```
}
user@host# show interfaces
ge-0/0/1 {
    unit 0 {
        family ethernet-switching {
            vlan {
                members 10;
            }
        }
    }
}
irb {
    unit 10 {
        family inet {
            address 10.1.1.10/24;
        }
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying VLANs

Purpose Verify that VLANs are configured and assigned to the interfaces.

Action From operational mode, enter the **show vlans** command.

```
user@host> show vlans

Routing instance        VLAN name             Tag          Interfaces
default-switch          default               1
                                                           ge-0/0/1.0
default-switch          v10                   10
                                                           ge-0/0/1.0
```

Meaning The output shows the VLAN is configured and assigned to the interface.

Related Documentation

- Understanding VLANs on page 769
- Ethernet Ports Switching Overview for Security Devices on page 706

Understanding VLAN Retagging on Security Devices

VLAN retagging is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D60.

Starting in Junos OS Release 15.1X49-D70, VLAN retagging in switching mode is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Starting in Junos OS Release 15.1X49-D80, VLAN retagging in switching mode is supported on SRX1500 devices.

The VLAN identifier in packets arriving on a Layer 2 trunk port can be rewritten or *retagged* with a different internal VLAN identifier. VLAN retagging is a symmetric operation; upon exiting the same trunk port, the retagged VLAN identifier is replaced with the original VLAN identifier. VLAN retagging provides a way to selectively screen incoming packets and redirect them to a firewall or other security device without affecting other VLAN traffic.

VLAN retagging can be applied only to interfaces configured as Layer 2 trunk interfaces. These interfaces can include redundant Ethernet interfaces in a Layer 2 transparent mode within a chassis cluster configuration.

NOTE: If a trunk port is configured for VLAN retagging, untagged packets received on the port are not assigned a VLAN identifier with the VLAN retagging configuration. To configure a VLAN identifier for untagged packets received on the physical interface, use the **native-vlan-id** statement.

To configure VLAN retagging for a Layer 2 trunk interface, specify a one-to-one mapping of the following:

- Incoming VLAN identifier—VLAN identifier of the incoming packet that is to be retagged. This VLAN identifier must not be the same VLAN identifier configured with the **native-vlan-id** statement for the trunk port.
- Internal VLAN identifier—VLAN identifier for the retagged packet. This VLAN identifier must be in the VLAN identifier list for the trunk port and must not be the same VLAN identifier configured with the **native-vlan-id** statement for the trunk port.

This is an enterprise style of VLAN retagging in which a single command **set interfaces ge-3/0/0 unit 0 family ethernet-switching vlan-rewrite translate 11 2** is sufficient on top of normal trunk configuration. But, in case of Q-in-Q which is is service provider style, the same thing can be done using swap.

Related Documentation

- Layer 2 Transparent Mode Overview on page 377
- Example: Configuring VLAN Retagging for Layer 2 Transparent Mode on a Security Device on page 384
- Example: Configuring Layer 2 Logical Interfaces on Security Devices on page 638

Configuring VLAN Retagging on a Layer 2 Trunk Interface of a Security Device

VLAN retagging is a feature that works on IEEE standard 802.1Q virtual LAN tagging (VLAN tagging. VLAN retagging for SRX1500 devices is an enterprise style of VLAN retagging, in which a single command is sufficient on top of normal trunk configuration.

1. Create a Layer 2 trunk interface.

```
[edit]
user@host#  set interfaces ge-3/0/0 unit 0 family ethernet-switching interface-mode trunk
vlan members 1–10
```

2. Configure VLAN retagging.

```
[edit]
user@host#  set interfaces ge-3/0/0 unit 0 family ethernet-switching vlan-rewrite translate
11 2
```

Related Documentation

- Understanding VLAN Retagging on Security Devices on page 773

Example: Configuring a Guest VLAN on a Security Device

This example shows how to configure a guest VLAN for limited network access or for Internet-only access to avoid compromising a company's security.

Guest VLANs are not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D60.

Requirements

Before you begin, verify that the interfaces that will be used are in switch mode. See "Example: Configuring Switching Modes on Security Devices" on page 713 and "Understanding Switching Modes on Security Devices" on page 705.

Overview

In this example, you configure a VLAN called visitor-vlan with a VLAN ID of 300. Then you set protocols and configure visitor-vlan as the guest VLAN.

Configuration

Step-by-Step Procedure

To configure a guest VLAN:

1. Configure a VLAN.

```
[edit]
user@host# set vlans visitor-vlan vlan-id 300
```

2. Specify the guest VLAN.

```
[edit]
user@host# set protocols dot1x authenticator interface all guest-vlan visitor-vlan
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Verification

To verify the configuration is working properly, enter the **show vlans** and **show protocols dot1x** commands.

Related Documentation

- Understanding VLANs on page 769
- Ethernet Ports Switching Overview for Security Devices on page 706

CHAPTER 33

Configuring Link Aggregation Control Protocol on Security Devices

Understanding Link Aggregation Control Protocol

LACP, a subcomponent of IEEE 802.3ad, provides additional functionality for link aggregation groups (LAGs). Use the link aggregation feature to aggregate one or more Ethernet interfaces to form a logical point-to-point link, known as a LAG, virtual link, or bundle. The MAC client can treat this virtual link like a single link.

Starting in Junos OS Release 15.1X49-D80, Link Aggregation Control Protocol (LACP) is supported in Layer 2 transparent mode in addition to existing support in Layer 3 mode for SRX300, SRX320, SRX340, SRX345, SRX1500, SRX4100, SRX4200 devices and vSRX instances. Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, Link Aggregation Control Protocol (LACP) is supported in Layer 2 transparent mode in addition to existing support in Layer 3 mode for SRX5400, SRX5600 and SRX5800 devices. When the SRX Series device uses LACP to bundle the member links, it creates high-speed connections, also known as *fat pipe*, with peer systems. Bandwidth can be increased by adding member links. Increased bandwidth is important especially for redundant Ethernet (reth) and aggregated Ethernet (ae) interfaces, for transmitting and receiving packets to and from the peer end for the whole system. LACP also provides automatic determination, configuration, and monitoring member links. LACP is compatible with other peers that run the 802.3ad LACP protocol. It automatically binds the member links without manually configuring the LAG, thereby avoiding errors.

NOTE: Tentative sessions are created for all interfaces in a particular VLAN. If there is plenty of one-way traffic, numerous tentative sessions are created. When sessions reach the maximum limit, vector fails and packet loss might be seen.

This topic contains the following sections:

Link Aggregation Benefits

Link aggregation increases bandwidth, provides graceful degradation as failure occurs, and increases availability. It provides network redundancy by load-balancing traffic across all available links. If one of the links fails, the system automatically load-balances traffic across all remaining links.

When LACP is not enabled, a local LAG might attempt to transmit packets to a remote single interface, which causes the communication to fail. When LACP is enabled, a local LAG cannot transmit packets unless a LAG with LACP is also configured on the remote end of the link.

A typical LAG deployment includes aggregate trunk links between an access switch and a distribution switch or customer edge (CE) device.

Link Aggregation Configuration Guidelines

When configuring link aggregation, note the following guidelines and restrictions:

- Link aggregation is supported only for Ethernet interfaces that are configured in switching mode (**family ethernet-switching**). Aggregating interfaces that are configured in routed mode (**family inet**) is also supported.
- You can configure a LAG by specifying the link number as a physical device and then associating a set of ports with the link. All the ports must have the same speed and be in full-duplex mode. Junos OS assigns a unique ID and port priority to each port. The ID and priority are not configurable.
- You can optionally configure LACP for link negotiation.
- You can optionally configure LACP for link protection.
- You can create up to eight Ethernet ports in each bundle.
- Each LAG must be configured on both sides of the link. The ports on either side of the link must be set to the same speed. At least one end of the LAG must be configured as active.
- LAGs are not supported on virtual chassis port links.
- By default, Ethernet links do not exchange protocol data units (PDUs), which contain information about the state of the link. You can configure Ethernet links to actively transmit PDUs, or you can configure the links to passively transmit them, sending out LACP PDUs only when they receive them from another link. The transmitting link is known as the actor and the receiving link is known as the partner.
- LAGs can only be used for a point-to-point connection.

For LACP configuration details, see Table 108 on page 779 and Table 109 on page 779.

Table 108: LACP (Link Aggregation Control Protocol) Configuration

Field	Function
Aggregated Interface	Indicates the name of the aggregated interface.
Link Status	Indicates whether the interface is linked (Up) or not linked (Down).
VLAN (VLAN ID)	Virtual LAN identifier value for IEEE 802.1Q VLAN tags (0.4094).
Description	The description for the LAG.

Table 109: Details of Aggregation

Field	Function
Administrative Status	Displays if the interface is enabled (Up) or disabled (Down).
Logical Interfaces	Shows the logical interface of the aggregated interface.
Member Interfaces	Member interfaces hold all the aggregated interfaces of the selected interfaces.
Port Mode	Specifies the mode of operation for the port: trunk or access.
Native VLAN (VLAN ID)	VLAN identifier to associate with untagged packets received on the interface.
IP Address/Subnet Mask	Specifies the address of the aggregated interfaces.
IPV6 Address/Subnet Mask	Specifies the IPV6 address of the aggregated interfaces.

For aggregated Ethernet interface options, see Table 110 on page 779.

Table 110: Aggregated Ethernet Interface Options

Field	Function	Action
Aggregated Interface	Indicates the name of the aggregated interface.	Enter the aggregated interface name. If an aggregated interface already exists, then the field is displayed as read-only.
LACP Mode	Specifies the mode in which LACP packets are exchanged between the interfaces. The modes are: • None—Indicates that no mode is applicable. • Active—Indicates that the interface initiates transmission of LACP packets • Passive—Indicates that the interface only responds to LACP packets.	Select from the list.

Table 110: Aggregated Ethernet Interface Options (continued)

Field	Function	Action
Description	The description for the LAG.	Enter the description.
Interface	Indicates that the interfaces available for aggregation.	Click **Add** to select the interfaces. NOTE: Only interfaces that are configured with the same speeds can be selected together for a LAG.
Speed	Indicates the speed of the interface.	
Enable Log	Specifies whether to enable generation of log entries for LAG.	Select to enable log generation.

NOTE: On SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345 and SRX650 devices, the speed mode and link mode configuration are available for member interfaces of ae. (Platform support depends on the Junos OS release in your installation.)

For VLAN options, see Table 111 on page 780.

Table 111: Edit VLAN Options

Field	Function	Action
Port Mode	Specifies the mode of operation for the port: trunk or access.	If you select Trunk, you can: 1. Click **Add** to add a VLAN member. 2. Select the VLAN and click **OK**. 3. (Optional) Associate a native VLAN ID with the port. If you select Access, you can: 1. Select the VLAN member to be associated with the port. 2. (Optional) Associate a VoIP VLAN with the interface. Only a VLAN with a VLAN ID can be associated as a VoIP VLAN. 3. Click **OK**.
VLAN Options	For trunk interfaces, the VLANs for which the interface can carry traffic.	Click **Add** to select VLAN members.
Native VLAN	VLAN identifier to associate with untagged packets received on the interface.	Select the VLAN identifier.

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS Release 15.1X49-D80, Link Aggregation Control Protocol (LACP) is supported in Layer 2 transparent mode in addition to existing support in Layer 3 mode for SRX300, SRX320, SRX340, SRX345, SRX1500, SRX4100, SRX4200 devices and vSRX instances.
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, Link Aggregation Control Protocol (LACP) is supported in Layer 2 transparent mode in addition to existing support in Layer 3 mode for SRX5400, SRX5600 and SRX5800 devices.

Related Documentation

- *Example: Configuring Link Aggregation Control Protocol on a Security Device (J-Web Procedure)*
- Ethernet Ports Switching Overview for Security Devices on page 706
- *Verifying Switching Mode Configuration*

Example: Configuring Link Aggregation Control Protocol on a Security Device (CLI Procedure)

This example shows how to configure LACP.

Requirements

This example uses an SRX Series device.

Before you begin:

- Determine which interfaces to use and verify that they are in switch mode. See "Understanding VLANs" on page 769.

Overview

In this example, for aggregated Ethernet interfaces, you configure the Link Aggregation Control Protocol (LACP). LACP is one method of bundling several physical interfaces to form one logical interface.

Configuration

CLI Quick Configuration

To quickly configure this section of the example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/6 ether-options 802.3ad ae0
set interfaces ge-0/0/7 ether-options 802.3ad ae0
set interfaces ae0 vlan-tagging
set interfaces ae0 aggregated-ether-options lacp active periodic fast
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set vlan vlan1000 vlan-id 1000
set interfaces ae0 unit 0 family ethernet-switching vlan members vlan1000
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure LACP:

1. Configure the interfaces for ae0.

   ```
   [edit ]
   user@host# set interfaces ge-0/0/6 ether-options 802.3ad ae0
   user@host# set interfaces ge-0/0/7 ether-options 802.3ad ae0
   ```

2. Configure ae0 interface for vlan tagging.

   ```
   [edit ]
   user@host# set interfaces ae0 vlan-tagging
   ```

3. Configure LACP for ae0 and configure periodic transmission of LACP packets.

   ```
   [edit ]
   user@host# set interfaces ae0 aggregated-ether-options lacp active periodic fast
   ```

4. Configure ae0 as a trunk port.

   ```
   [edit ]
   user@host# set interfaces ae0 unit 0 family ethernet-switching interface-mode
     trunk
   ```

5. Configure the VLAN.

   ```
   [edit ]
   user@host# set vlan vlan1000 vlan-id 1000
   ```

6. Add the ae0 interface to the VLAN.

   ```
   [edit ]
   user@host# set interfaces ae0 unit 0 family ethernet-switching vlan members
     vlan1000
   ```

7. If you are done configuring the device, commit the configuration.

   ```
   [edit ]
   user@host# commit
   ```

Results From configuration mode, confirm your configuration by entering the **show interfaces** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show interfaces
ge-0/0/6 {
   ether-options {
      802.3ad ae0;
   }
}
ge-0/0/7 {
   ether-options {
      802.3ad ae0;
   }
}
ae0 {
   vlan- tagging;
   aggregated-ether-options {
      lacp {
         active;
         periodic fast;
      }
   }
   unit 0 {
      family ethernet-switching {
         interface-mode trunk;
         vlan {
            members vlan1000;
         }
      }
   }
}
```

Verification

Verifying LACP Statistics

Purpose Display LACP statistics for aggregated Ethernet interfaces.

Action From operational mode, enter the **show lacp statistics interfaces ae0** command.

```
user@host> show lacp statistics interfaces ae0
Aggregated interface: ae0
    LACP Statistics:       LACP Rx     LACP Tx   Unknown Rx   Illegal Rx
      ge-0/0/6                1352        2035            0            0
      ge-0/0/7                1352        2056            0            0
```

Meaning The output shows LACP statistics for each physical interface associated with the aggregated Ethernet interface, such as the following:

- The LACP received counter that increments for each normal hello packet received
- The number of LACP transmit packet errors logged
- The number of unrecognized packet errors logged
- The number of invalid packets received

Use the following command to clear the statistics and see only new changes:

```
user@host# clear lacp statistics interfaces ae0
```

Verifying LACP Aggregated Ethernet Interfaces

Purpose Display LACP status information for aggregated Ethernet interfaces.

Action From operational mode, enter the **show lacp interfaces ae0** command.

```
user@host> show lacp interfaces ae0
Aggregated interface: ae0
    LACP state:       Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
      ge-0/0/6       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-0/0/6     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Passive

      ge-0/0/7       Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
      ge-0/0/7     Partner    No    No   Yes  Yes  Yes   Yes     Fast    Passive

    LACP protocol:        Receive State  Transmit State          Mux State
      ge-0/0/6                  Current   Fast periodic Collecting distributing
      ge-0/0/7                  Current   Fast periodic Collecting distributing
```

Meaning The output shows aggregated Ethernet interface information, including the following information:

- The LACP state—Indicates whether the link in the bundle is an actor (local or near-end of the link) or a partner (remote or far-end of the link).
- The LACP mode—Indicates whether both ends of the aggregated Ethernet interface are enabled (active or passive)—at least one end of the bundle must be active.
- The periodic link aggregation control PDU transmit rate.
- The LACP protocol state—Indicates the link is up if it is collecting and distributing packets.

Related Documentation

- Understanding Link Aggregation Control Protocol on page 777
- Ethernet Ports Switching Overview for Security Devices on page 706

Example: Configuring Aggregated Ethernet Device with LAG and LACP on a Security Device (CLI Procedure)

Requirements

No special configuration beyond device initialization is required before configuring this feature.

Overview

This example shows the configuration of aggregated Ethernet (ae) devices with LAG and LACP.

Configuration

Step-by-Step Procedure

To configure LAG:

1. Configure the number of aggregated Ethernet interfaces with LAG interface that you need to create. Set the device-count option to 5.

   ```
   [edit]
   user@host# set chassis aggregated-devices ethernet device-count 5
   ```

2. Add a port to the aggregated Ethernet interface with LAG.

   ```
   [edit]
   user@host# set interfaces ge-2/0/1 ether-options 802.3ad ae0
   user@host# set interfaces ge-2/0/2 ether-options 802.3ad ae0
   ```

3. Configure LACP for the aggregated Ethernet interface with LAG.

   ```
   [edit]
   user@host# set interfaces ae0 aggregated-ether-options lacp active
   ```

4. Configure family Ethernet switching for the aggregated Ethernet interface with LAG.

   ```
   [edit]
   user@host# set interfaces ae0 unit 0 family ethernet-switching
   ```

5. Configure the VLAN vlan20 with VLAN ID 20.

   ```
   [edit]
   user@host# set vlans vlan20 vlan-id 20
   ```

6. Add the aggregated Ethernet interface to the VLAN.

```
[edit]
user@host# set vlans vlan20 interface ae0
```

7. Check the configuration by entering the **show vlans** and **show interfaces** commands

```
user@host# show vlans
vlan20 {
    vlan-id 20;
    interface {
        ae0.0;
    }
}
```

```
user@host# show interfaces
ge-2/0/1 {
    ether-options {
        802.3ad ae0;
    }
}
ge-2/0/2 {
    ether-options {
        802.3ad ae0;
    }
}
ae0 {
    aggregated-ether-options {
        lacp {
            active;
        }
    }
    unit 0 {
        family ethernet-switching;
    }
}
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

NOTE: Likewise, you can configure other devices with LAG and LACP.

Verification

Verifying Aggregated Ethernet Interface with LAG and LACP

Purpose Verify that you can configure aggregated Ethernet interfaces with LAG and LACP.

Action From configuration mode, enter the **show lacp interfaces** to view the LACP interfaces.

```
user@host# run show lacp interfaces
Aggregated interface: ae0
    LACP state:       Role   Exp   Def  Dist  Col  Syn  Aggr  Timeout  Activity
```

```
    ge-2/0/1     Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
    ge-2/0/1   Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
    ge-2/0/2     Actor    No    No   Yes  Yes  Yes   Yes     Fast    Active
    ge-2/0/2   Partner    No    No   Yes  Yes  Yes   Yes     Fast    Active
  LACP protocol:        Receive State  Transmit State          Mux State
    ge-2/0/1               Current   Fast periodic Collecting distributing
    ge-2/0/2               Current   Fast periodic Collecting distributing
```

From configuration mode, enter the **show vlans** command to view the VLAN interfaces.

```
user@host# run show vlans
Name           Tag     Interfaces
default         1      None
vlan20          20     ae0.0
```

From configuration mode, enter the **show interfaces (interface name)** command to view the status of the ge-2/0/1 and ge-2/0/2 interfaces.

```
user@host# run show interfaces ge-2/0/1 terse
Interface               Admin Link Proto    Local                 Remote
ge-2/0/1                up    up
ge-2/0/1.0              up    up   aenet    --> ae0.0
```

```
user@host# run show interfaces ge-2/0/2 terse
Interface               Admin Link Proto    Local                 Remote
ge-2/0/2                up    up
ge-2/0/2.0              up    up   aenet    --> ae0.0
```

Meaning The output shows the aggregated Ethernet Interface with LAG and LACP is configured.

Related Documentation

- *Understanding Aggregated Ethernet Interfaces*
- *Understanding LACP on Standalone Devices*
- *Example: Configuring Link Aggregation Control Protocol (CLI Procedure)*

CHAPTER 34

Configuring 802.1X Port-Based Network Authentication on Security Devices

Understanding 802.1X Port-Based Network Authentication

NOTE: From Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D75 and Junos OS Release 17.3R1, IEEE 802.1X port-based network authentication is not supported.

NOTE: Starting in Junos OS 15.1X49-D80, 802.1X port-based authentication is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

Both IEEE 802.1X authentication and MAC RADIUS authentication both provide network edge security, protecting Ethernet LANs from unauthorized user access by blocking all traffic to and from devices at the interface until the supplicant's credential or MAC address is presented and matched on the *authentication server* (a RADIUS server). When the supplicant is authenticated, the switch stops blocking access and opens the interface to the supplicant.

A LAN network configured for 802.1X authentication contains three basic components:

- **Supplicant**—The IEEE term for a host that requests to join the network. The host can be responsive or nonresponsive. A responsive host is one on which 802.1X authentication is enabled and that provides authentication credentials (such as a user name and password). A nonresponsive host is one on which 802.1X authentication is not enabled.
- **Authenticator port access entity**—The IEEE term for the authenticator. The SRX Series device is the authenticator and controls access by blocking all traffic from host/supplicant until they are authenticated.

- **Authentication server**—The server containing the back-end database that makes authentication decisions. (Junos OS supports RADIUS authentication servers.) The authentication server contains credential information for each supplicant that can connect to the network. The authenticator forwards credentials supplied by the supplicant to the authentication server. If the credentials forwarded by the authenticator match the credentials in the authentication server database, access is granted. If the credentials forwarded do not match, access is denied.

Table 112 on page 790 lists the features that the implementation of 802.1X authentication provides for specific devices. (Platform support depends on the Junos OS release in your installation.). Table 113 on page 790 lists the supplicant capacities that the implementation of 802.1X authentication provides for specific devices.

Table 112: 802.1X Authentication Features

Feature	SRX300/SRX320	SRX340/SRX345	SRX550M	SRX1500
Dynamic VLAN assignment	Yes	Yes	Yes	Yes
MAC RADIUS authentication	Yes	Yes	Yes	Yes
Static MAC bypass	Yes	Yes	Yes	Yes
Guest VLAN	Yes	Yes	Yes	Yes
RADIUS server failure fallback	Yes	Yes	Yes	Yes
VoIP VLAN support	Yes	Yes	Yes	Yes
RADIUS accounting	Yes	Yes	Yes	Yes

Table 113: 802.1x Supplicant Capacities

Capacities	SRX300/SRX320	SRX340/SRX345	SRX550M	SRX1500
Supplicants per port	64	64	64	64
Supplicants per system	2K	2K	2K	2K
Supplicants with dynamic VLAN assignments	64	300	2K	2K

This topic contains the following sections:

Dynamic VLAN Assignment

When a supplicant first connects to an SRX Series device, the authenticator sends a request to the supplicant to begin 802.1X authentication. If the supplicant is an 802.1X-enabled device, it responds, and the authenticator relays an authentication request to the RADIUS server.

As part of the reply to the authentication request, the RADIUS server returns information about the VLAN to which the port belongs. By configuring the VLAN information at the RADIUS server, you can control the VLAN assignment on the port.

MAC RADIUS Authentication

If the authenticator sends three requests to a supplicant to begin 802.1X authentication and receives no response, the supplicant is considered nonresponsive. For a nonresponsive supplicant, the authenticator sends a request to the RADIUS server for authentication of the supplicant's MAC address. If the MAC address matches an entry in a predefined list of MAC addresses on the RADIUS server, authentication is granted and the authenticator opens LAN access on the interface where the supplicant is connected.

You can configure the number of times the authenticator attempts to receive a response and the time period between attempts.

Static MAC Bypass

The authenticator can allow particular supplicants direct access to the LAN, bypassing the authentication server, by including the supplicants' MAC addresses in the static MAC bypass list configured on the SRX Series device. Supplicants' MAC addresses are first checked against this list. If a match is found, the corresponding supplicant is considered successfully authenticated and the interface is opened up for it. No further authentication is done for that supplicant. If a match is not found and 802.1X authentication is enabled for the supplicant, the device continues with MAC RADIUS authentication on the authentication server.

For each MAC address in the list, you can configure the VLAN to which the supplicant is moved or the interfaces on which the supplicant can connect.

Guest VLAN

You can specify a guest VLAN that provides limited network access for nonresponsive supplicants. If a guest VLAN is configured, the authenticator connects all nonresponsive supplicants to the predetermined VLAN, providing limited network access, often only to the Internet. This type of configuration can be used to provide Internet access to visitors without compromising company security.

NOTE: In 802.1X, MAC RADIUS, and guest VLAN must not be configured together, because guest VLAN does not work when MAC RADIUS is configured.

IEEE 802.1X provides LAN access to nonresponsive hosts, which are hosts where 802.1X is not enabled. These hosts, referred to as guests, typically are provided access only to the Internet.

RADIUS Server Failure Fallback

You can define one of four actions to be taken if no RADIUS authentication server is reachable (if, for example, a server failure or a timeout has occurred on the authentication server).

- **deny**—(default) Prevent traffic from flowing from the supplicant through the interface.
- **permit**—Allow traffic to flow from the supplicant through the interface as if the supplicant were successfully authenticated by the RADIUS server.
- **use-cache**—Force successful authentication if authentication was granted before the failure or timeout. This ensures that authenticated users are not adversely affected by a failure or timeout.
- **vlan *vlan-name* | *vlan-id*** —Move the supplicant to a different VLAN specified by name or ID. This applies only to the first supplicant connecting to the interface.

NOTE: For the permit, use-cache, and vlan fallback actions to work, 802.1X supplicants need to accept an out-of-sequence SUCCESS packet.

For RADIUS server settings, see Table 114 on page 792.

Table 114: RADIUS Server Settings

Field	Function	Your Action
IP Address	Specifies the IP address of the server.	Enter the IP address in dotted decimal notation.
Password	Specifies the login password.	Enter the password.
Confirm Password	Verifies the login password for the server.	Reenter the password.
Server Port Number	Specifies the port with which the server is associated.	Type the port number.
Source Address	Specifies the source address of the SRX Series device for communicating with the server.	Type the IP address in dotted decimal notation.
Retry Attempts	Specifies the number of login retries allowed after a login failure.	Type the number.

Table 114: RADIUS Server Settings (continued)

Field	Function	Your Action
Timeout	Specifies the time interval to wait before the connection to the server is closed.	Type the interval in seconds.

For 802.1X exclusion list details, see Table 115 on page 793.

Table 115: 802.1X Exclusion List

Field	Function	Your Action
MAC Address	Specifies the MAC address to be excluded from 802.1X authentication.	Enter the MAC address.
Exclude if connected through the port	Specifies that a supplicant can bypass authentication if it is connected through a particular interface.	Select to enable the option. Select the port through which the supplicant is connected.
Move the host to the VLAN	Moves the host to a specific VLAN once the host is authenticated.	Select to enable the option. Select the VLAN from the list.

For 802.1X port settings, see Table 116 on page 793.

Table 116: 802.1X Port Settings

Field	Function	Your Action
Supplicant Mode		
Supplicant Mode	Specifies the mode to be adopted for supplicants: • Single secure—Allows only one host for authentication. • Multiple—Allows multiple hosts for authentication. Each host is checked before being admitted to the network. • Single mode authentication for multiple hosts—Allows multiple hosts but only the first is authenticated.	Select the required mode.
Authentication		
Enable re-authentication	Specifies enabling reauthentication on the selected interface.	Select to enable reauthentication. Enter the timeout for reauthentication in seconds.
Action for nonresponsive hosts	Specifies the action to be taken in case a supplicant is nonresponsive: • Move to the Guest VLAN—Moves the supplicant to the specified Guest VLAN. • Deny—Does not permit access to the supplicant.	Select the required action.

Table 116: 802.1X Port Settings (continued)

Field	Function	Your Action
Timeouts	Specifies timeout values for: • Port waiting time after an authentication failure • EAPOL retransmitting interval • Maximum EAPOL requests • Maximum number of retries • Port timeout value for a response from the supplicant • Port timeout value for a response from the RADIUS server	Enter timeout values in seconds for the appropriate options.

VoIP VLAN Support

When VoIP is used with 802.1X, the RADIUS server authenticates the phone, and Link Layer Discovery Protocol–Media Endpoint Discovery (LLDP-MED) provides the class-of-service (CoS) parameters for the phone.

You can configure 802.1X authentication to work with VoIP in multiple-supplicant or single-supplicant mode:

- **Multiple-supplicant mode**—Allows multiple supplicants to connect to the interface. Each supplicant is authenticated individually.
- **Single-supplicant mode**—Authenticates only the first supplicant. All other supplicants that connect later to the interface are allowed to *piggyback* on the first supplicant's authentication and gain full access.

RADIUS Accounting

Configuring RADIUS accounting on a SRX Series device lets you collect statistical data about users logging in to and out off a LAN, and sends it to a RADIUS accounting server. The collected data can be used for general network monitoring, to analyze and track usage patterns, or to bill a user on the basis of the amount of time or type of services accessed.

To configure RADIUS accounting, specify one or more RADIUS accounting servers to receive the statistical data from the device, and select the type of accounting data to be collected. To view the collected statistics, you can access the log file configured to receive them.

Server Reject VLAN

By default, when authentication fails, the supplicant is denied access to the network. However, you can specify a VLAN to which the supplicant is moved if authentication fails. The server reject VLAN is similar to a guest VLAN. With a server reject VLAN, however, authentication is first attempted by credential, then by MAC address. If both authentication methods fail, the supplicant is given access to a predetermined VLAN with limited network access.

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS 15.1X49-D80, 802.1X port-based authentication is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.
15.1X49-D40	From Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D75 and Junos OS Release 17.3R1, IEEE 802.1X port-based network authentication is not supported.

Related Documentation

- Ethernet Ports Switching Overview for Security Devices on page 706
- *Verifying Switching Mode Configuration*

Example: Specifying RADIUS Server Connections on a Security Device

This example shows how to specify a RADIUS server for 802.1X authentication to provide network edge security.

NOTE: From Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D75 and Junos OS Release 17.3R1, specifying a RADIUS server for 802.1X authentication is not supported.

NOTE: Starting in Junos OS 15.1X49-D80, 802.1X port-based authentication is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

Requirements

Before you begin, verify that the interfaces used are in switch mode. See "Example: Configuring Switching Modes on Security Devices" on page 713.

- To use 802.1X or MAC RADIUS authentication, you must specify the connections on the SRX Series device for each RADIUS server to which you will connect.

Overview

In this example, you set the RADIUS server IP address to 10.204.96.165 and the secret password to abc. The secret password on the device must match the secret password

on the server. You can set the number of retries after which port is placed into wait state to 5.

Then you create a profile called **profile1** and set the authentication order to **radius**. You can specify one or more RADIUS servers to be associated with **profile1**. Finally, you define profile1 as the authentication profile for 802.1X or MAC RADIUS authenticator.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set access radius-server 10.204.96.165 secret abc
set access radius-server 10.204.96.165 retry 5
set access profile profile1 authentication-order radius
set access profile profile1 radius authentication-server 10.204.96.165
set access profile profile1 radius accounting-server 10.204.96.165
set protocols dot1x authenticator authentication-profile-name profile1
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To specify a RADIUS server for 802.1X authentication:

1. Configure access.

   ```
   [edit]
   user@host# edit access
   ```

   NOTE: For 802.1X authentication, the RADIUS server must be configured at the access hierarchy level.

2. Define the IP address and the secret password for the RADIUS server.

   ```
   [edit access]
   user@host# set access radius-server 10.204.96.165 secret abc
   ```

3. Specify the number of retries after which port is placed into wait state to 5.

   ```
   [edit access]
   user@host# set access radius-server 10.204.96.165 retry 5
   ```

4. Create the profile.

   ```
   [edit access]
   user@host# edit profile profile1
   ```

5. Configure the authentication order.

```
[edit access profile profile1]
user@host# set authentication-order radius
```

6. Specify one or more RADIUS servers to be associated with profile1.

```
[edit access profile profile1]
user@host# set radius authentication-server 10.204.96.165
user@host# set radius accounting-server 10.204.96.165
```

7. Define authentication profile.

```
[edit]
user@host# set protocols dot1x authenticator authentication-profile-name profile1
```

Results From configuration mode, confirm your configuration by entering the **show access** and **show protocols dot1x** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show access
  radius-server {
    10.204.96.165 {
      secret "$ABC123"; ## SECRET-DATA
      retry 5;
    }
  }
  profile profile1 {
    authentication-order radius;
    radius {
    authentication-server 10.204.96.165;
    accounting-server 10.204.96.165;
  }
}
[edit]
  user@host# show protocols dot1x
  authenticator {
    interface {
      ge-0/0/0.0 {
        supplicant multiple
        mac-radius;
        no-reauthentication;
        server-fail permit;
      }
      ge-0/0/1.0 {
        supplicant multiple
        mac-radius;
        no-reauthentication;
      }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying a RADIUS Server

Purpose Verify that the RADIUS server is configured properly.

Action From configuration mode, enter the **show access** and **show protocols dot1x** commands.

```
user@host# show access
  radius-server {
    10.204.96.165 {
      secret "$ABC123"; ## SECRET-DATA
      retry 5;
    }
  }
  profile profile1 {
    authentication-order radius;
    radius {
    authentication-server 10.204.96.165;
    accounting-server 10.204.96.165;
  }
}
user@host# show protocols dot1x
authenticator {
    static {
      00:50:56:85:66:0f/48 {
        vlan-assignment vlan6;
        interface ge-0/0/0.0;
      }
      00:50:56:9e:56:42/48 {
        vlan-assignment vlan6;
        interface ge-0/0/1.0;
      }
  }
    interface {
      ge-0/0/0.0 {
        supplicant multiple;
        server-fail deny;
      }
      ge-0/0/1.0 {
        supplicant multiple;
        server-fail deny;
      }
  }
l2-learning {
  global-mac-table-aging-time 60;
  global-mode switching;
}
```

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS 15.1X49-D80, 802.1X port-based authentication is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.
15.1X49-D40	From Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D75 and Junos OS Release 17.3R1, specifying a RADIUS server for 802.1X authentication is not supported.

Related Documentation

- Ethernet Ports Switching Overview for Security Devices on page 706
- Understanding 802.1X Port-Based Network Authentication on page 789
- Understanding Switching Modes on Security Devices on page 705
- Understanding VLANs on page 769

Example: Configuring 802.1X Interface Settings on a Security Device

This example shows how to configure 802.1X interface settings for network edge security.

NOTE: From Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D75 and Junos OS Release 17.3R1, configuring 802.1X port-based authentication interface settings is not supported.

NOTE: Starting in Junos OS 15.1X49-D80, 802.1X port-based authentication is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

Requirements

Before you begin:

- Verify that the interfaces used are in switch mode. See "Example: Configuring Switching Modes on Security Devices" on page 713.
- Ensure that the interfaces are defined in the interfaces hierarchy with family ethernet-switching.

Overview

In this example, you set the supplicant mode to **multiple** after configuring protocol **dot1x** and authenticator interface ge-0/0/0.0. You then enable reauthentication and set the reauthentication interval to **120**. You then configure the timeout for the interface before it resends an authentication request to the RADIUS server as **5**. You specify the time, in seconds, the interface waits before retransmitting the initial EAPoL PDUs to the supplicant as **60**. Set the **server-fail** to **deny** so that the server does not fail. Finally, you configure the maximum number of times an EAPoL request packet is retransmitted to the supplicant before the authentication session times out as **5**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols dot1x authenticator interface ge-0/0/0.0 supplicant multiple
  reauthentication 120
set protocols dot1x authenticator interface ge-0/0/0.0 server-timeout 5 transmit-period
  60
set protocols dot1x authenticator interface ge-0/0/0.0 server-fail deny
set protocols dot1x authenticator interface ge-0/0/0.0 maximum-requests 5
set protocols dot1x authenticator interface ge-0/0/2.0 supplicant multiple
  reauthentication 120
set protocols dot1x authenticator interface ge-0/0/2.0 server-timeout 5 transmit-period
  60
set protocols dot1x authenticator interface ge-0/0/2.0 server-fail deny
set protocols dot1x authenticator interface ge-0/0/2.0 maximum-requests 5
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To specify a RADIUS server for 802.1X authentication:

1. Configure the protocol.

   ```
   [edit]
   user@host# set protocols dot1x
   ```

2. Configure an interface.

   ```
   [edit protocols dot1x]
   user@host# set authenticator interface ge-0/0/0.0
   ```

3. Configure the supplicant mode.

   ```
   [edit protocols dot1x authenticator interface ge-0/0/0.0]
   user@host# set supplicant multiple
   ```

4. Enable reauthentication and specify the reauthentication interval.

```
[edit protocols dot1x authenticator interface ge-0/0/0.0]
user@host# set reauthentication 120
```

5. Configure and set the server timeout value.

```
[edit protocols dot1x authenticator interface ge-0/0/0.0]
user@host# set server-timeout 5
```

6. Configure transmit period.

```
[edit protocols dot1x authenticator interface ge-0/0/0.0]
user@host# set transmit-period 60
```

7. Set **server-fail** to **deny**

```
[edit protocols dot1x authenticator interface ge-0/0/0.0]
user@host# set server-fail deny
```

8. Specify the maximum requests value.

```
[edit protocols dot1x authenticator interface ge-0/0/0.0]
user@host# set maximum-requests 5
```

Results From configuration mode, confirm your configuration by entering the **show protocols dot1x** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show protocols dot1x
  authenticator {
    interface {
      ge-0/0/0.0 {
        supplicant multiple;
        transmit-period 60;
        mac-radius;
        reauthentication 120;
        server-timeout 5;
        maximum-requests 5;
        server-fail permit;
      }
      ge-0/0/1.0 {
        supplicant multiple;
        mac-radius;
        no-reauthentication;
      }
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Verifying 802.1X Interface Settings

Purpose Verify that the 802.1X interface settings are working properly.

Action From configuration mode, enter the **show protocols dot1x** command.

```
user@host# show protocols dot1x
authenticator {
    interface {
        ge-0/0/0.0 {
            supplicant multiple;
            server-fail deny;
        }
        ge-0/0/1.0 {
            supplicant multiple;
            server-fail deny;
        }
    }
```

Release History Table

Release	Description
15.1X49-D80	Starting in Junos OS 15.1X49-D80, 802.1X port-based authentication is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.
15.1X49-D40	From Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D75 and Junos OS Release 17.3R1, configuring 802.1X port-based authentication interface settings is not supported.

Related Documentation

- Ethernet Ports Switching Overview for Security Devices on page 706
- Understanding 802.1X Port-Based Network Authentication on page 789
- Understanding Switching Modes on Security Devices on page 705
- Understanding VLANs on page 769

CHAPTER 35

Configuring Port Security on Security Devices

Port Security Overview

Ethernet LANs are vulnerable to attacks such as address spoofing (forging) and Layer 2 denial of service (DoS) attacks on network devices. Port security features help protect the access ports on your services gateway against the losses of information and productivity that can result from such attacks.

Junos OS on SRX Series devices provides features to help secure ports on a switching port on the services gateway. The ports can be categorized as either trusted or untrusted. You apply policies appropriate to those categories to protect against various types of attacks.

The MAC limit port security feature can be turned on to obtain the most robust port security level. Basic port security features are enabled in the services gateway's default configuration. You can configure additional features with minimal configuration steps.

Related Documentation

- Ethernet Ports Switching Overview for Security Devices on page 706
- Understanding MAC Limiting on page 803
- *Verifying Switching Mode Configuration*

Understanding MAC Limiting

MAC limiting protects against flooding of the Ethernet switching table (also known as the MAC forwarding table or Layer 2 forwarding table). You enable this feature on interfaces (ports).

MAC limiting sets a limit on the number of MAC addresses that can be learned dynamically on a single Layer 2 access interface or on all the Layer 2 access interfaces on the services gateway.

You configure the maximum number of dynamic MAC addresses allowed per interface. When the limit is exceeded, incoming packets with new MAC addresses are treated as specified by the configuration.

Starting with Junos OS Release 18.2R1, on SRX4100 and SRX4200 Series devices, the maximum range of MAC addresses configured on the VLAN interface is changed from 1 through 16383 to 1 through 5120. The short description of **interface-mac-limit** at the CLI command hierarchy is changed from **Maximum number of MAC addresses per interface (1..16383)** to **Maximum number of MAC addresses per interface (1..5120)** at the **[edit vlans vlan-name switch-options]** hierarchy level. Prior to Junos OS 18.2R1 Release, if you configure with the 16383 value, commit operation fails during commit.

You can choose to have one of the following actions performed when the MAC addresses limit is exceeded:

NOTE: Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the **log**, **none**, and **shutdown** actions are not supported.

- **drop**—Drop the packet and generate an alarm, an SNMP trap, or a system log entry. This is the default.
- **log**—Do not drop the packet but generate an alarm, an SNMP trap, or a system log entry.
- **none**—Take no action.
- **shutdown**—Disable the interface and generate an alarm. If you have configured the services gateway with the **port-error-disable** statement, the disabled interface recovers automatically upon expiration of the specified disable timeout. If you have not configured the services gateway for autorecovery from port error disabled conditions, you can bring up the disabled interfaces with running the **clear ethernet-switching recovery-timeout** command.

NOTE: MAC limit is applied only to new MAC learning requests. If you already have 10 learned MAC addresses and you configure the limit as 5, all the MACs will remain in the forwarding database (FDB) table. When the learned MAC addresses age out (or are cleared by the user with the clear ethernet-switching command), they are not relearned.

MAC limiting does not apply to static MAC addresses. Users can configure any number of static MAC addresses independent of MAC limiting and all of them are added to FDB.

NOTE: Starting with Junos OS Release 15.1X49-D60 and Junos OS Release 17.3R1, the maximum number of MAC addresses learned on all logical interfaces on the SRX1500 device is 24,575. When this limit is reached, incoming packets with a new source MAC address will be dropped.

Release History Table

Release	Description
18.2R1	Starting with Junos OS Release 18.2R1, on SRX4100 and SRX4200 Series devices, the maximum range of MAC addresses configured on the VLAN interface is changed from 1 through 16383 to 1 through 5120. The short description of **interface-mac-limit** at the CLI command hierarchy is changed from **Maximum number of MAC addresses per interface (1..16383)** to **Maximum number of MAC addresses per interface (1..5120)** at the **[edit vlans vlan-name switch-options]** hierarchy level. Prior to Junos OS 18.2R1 Release, if you configure with the 16383 value, commit operation fails during commit.
15.1X49-D40	Starting in Junos OS Release 15.1X49-D40 and Junos OS Release 17.3R1, the **log**, **none**, and **shutdown** actions are not supported.

Related Documentation

- Example: Configuring MAC Limiting on a Security Device on page 805
- Port Security Overview on page 803
- Ethernet Ports Switching Overview for Security Devices on page 706
- *Verifying Switching Mode Configuration*

Example: Configuring MAC Limiting on a Security Device

This example shows how to configure port security features by setting a MAC limit of 5.

Requirements

Before you begin, verify that the interfaces that will be used are in switch mode. See "Example: Configuring Switching Modes on Security Devices" on page 713 and "Understanding Switching Modes on Security Devices" on page 705.

Overview

MAC limiting protects against flooding of the Ethernet switching table on the SRX Series Services Gateways. MAC limiting sets a limit on the number of MAC addresses that can be learned on a single Layer 2 access interface (port).

This example shows how to configure port security features by setting a MAC limit of 5.

CLI Quick Configuration

To quickly configure this example, copy the following command, paste it into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the command into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set switch-options interface ge-0/0/1 interface-mac-limit 5
set interface ge–0/0/2 ether-options source-address-filter 00:00:5E:00:AA
set interface ge–0/0/2 ether-options source-address-filter 00:00:5E:00:AB
set interface ge–0/0/2 ether-options source-address-filter 00:00:5E:00:AC
```

Configuration

Step-by-Step Procedure

The action is not specified, so that the device performs the default action **drop** if the limit is exceeded:

NOTE: Do not set the mac-limit to 1. The first learned MAC address is often inserted into the FDB automatically (for example, for routed VLAN interfaces the first MAC address inserted into the forwarding database is the MAC address of the RVI; for Aggregated Ethernet bundles using LACP, the first MAC address inserted into the FDB in the forwarding table is the source address of the protocol packet). The services gateway will therefore not learn MAC addresses other than the automatic addresses when the mac-limit is set to 1, and this will cause problems with MAC learning and forwarding.

1. On a single interface (here, the interface is ge-0/0/1):

   ```
   [edit switch-options]
   user@host# set switch-options interface ge-0/0/1 interface-mac-limit 5
   ```

2. For specifying specific MAC addresses:

   - On a single interface (here, the interface is ge-0/0/2):

     ```
     [edit interfaces ether-options source-address-filter ]
     user@host# set interface ge–0/0/2 ether-options source-address-filter
       00:00:5E:00:AA
     user@host# set interface ge–0/0/2 ether-options source-address-filter
       00:00:5E:00:AB
     user@host# set interface ge–0/0/2 ether-options source-address-filter
       00:00:5E:00:AC
     ```

3. Enter **commit** from configuration mode.

Verification

Verifying That MAC Limiting Is Working Correctly on the Services Gateway

Purpose Verify that MAC limiting is working on the services gateway.

Action Display the learned MAC addresses. The following sample output shows the results when two packets were sent from hosts on ge-0/0/1 and five packets requests were sent from hosts on ge-0/0/2, with both interfaces set to a MAC limit of 4 with the action drop:

```
user@host> show ethernet-switching table
Ethernet-switching table: 7 entries, 6 learned
VLAN MAC address Type Age Interfaces
employee-vlan * Flood - ge-0/0/2.0
employee-vlan 00:00:5E:00:00 Learn 0 ge-0/0/1.0
employee-vlan 00:00:5E:00:AA Learn 0 ge-0/0/1.0
employee-vlan 00:00:5E:00:AB Learn 0 ge-0/0/2.0
employee-vlan 00:00:5E:00:AC Learn 0 ge-0/0/2.0
employee-vlan 00:00:5E:00:AD Learn 0 ge-0/0/2.0
employee-vlan 00:00:5E:00:AE Learn 0 ge-0/0/2.0
```

Meaning The sample output shows that with a MAC limit of 4 for each interface, the packet for a fifth MAC address on ge-0/0/2 was dropped because it exceeded the MAC limit. The address was not learned, and thus an asterisk (*) rather than an address appears in the MAC address column in the first line of the sample output.

Related Documentation

- Understanding MAC Limiting on page 803
- Ethernet Ports Switching Overview for Security Devices on page 706
- *Verifying Switching Mode Configuration*

Configuring Autorecovery From the Disabled State on Secure Interfaces of a Security Device (CLI Procedure)

An Ethernet switching access interface on a SRX Series device might shut down or be disabled as a result of one of the following port-security configurations:

- MAC limiting—**mac-limit** statement is configured with action **shutdown**.

You can configure the device to automatically restore the disabled interfaces to service after a specified period of time. Autorecovery applies to all the interfaces that have been disabled due to MAC limiting errors.

NOTE: You must specify the disable timeout value for the interfaces to recover automatically. There is no default disable timeout. If you do not specify a timeout value, you need to use the clear ethernet-switching recovery-timeout command to clear the errors and restore the interfaces or the specified interface to service.

To configure autorecovery from the disabled state due to MAC limiting shutdown actions:

```
[edit interfaces]
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching recovery-timeout 60
```

Related Documentation

- Understanding MAC Limiting on page 803
- Example: Configuring MAC Limiting on a Security Device on page 805
- clear ethernet-switching recovery-timeout on page 1172

CHAPTER 36

Configuring Ethernet OAM Connectivity Fault Management on Security Devices

Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways

Starting in Junos OS Release 15.1X49-D70, Ethernet OAM link fault management for SRX Series services gateways is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

The Ethernet interfaces on SRX Series devices support the IEEE 802.3ah standard for Operation, Administration, and Maintenance (OAM). The standard defines OAM link fault management (LFM). You can configure IEEE 802.3ah OAM LFM on point-to-point Ethernet links that are connected either directly or through Ethernet repeaters. The IEEE 802.3ah standard meets the requirement for OAM capabilities as Ethernet moves from being solely an enterprise technology to a WAN and access technology, and the standard remains backward-compatible with existing Ethernet technology.

The following OAM LFM features are supported:

- Discovery and link monitoring—The discovery process is triggered automatically when OAM is enabled on the interface. The discovery process permits Ethernet interfaces to discover and monitor the peer on the link if it also supports the IEEE 802.3ah standard. In active mode, the interface discovers and monitors the peer on the link if the peer also supports IEEE 802.3ah OAM functionality. In passive mode, the peer initiates the discovery process. After the discovery process has been initiated, both sides participate in discovery. The device performs link monitoring by sending periodic OAM protocol data units (PDUs) to advertise OAM mode, configuration, and capabilities.

  You can specify the number of OAM PDUs that an interface can miss before the link between peers is considered down.

- Remote fault detection—Remote fault detection uses flags and events. Flags convey Link Fault (a loss of signal), Dying Gasp (an unrecoverable condition such as a power failure), and Critical Event (an unspecified vendor-specific critical event). You can specify the periodic OAM PDU sending interval for fault detection. SRX Series devices use the Event Notification OAM PDU to notify the remote OAM device when a problem is detected. You can specify the action to be taken by the system when the configured link-fault event occurs.
- Remote loopback—Remote loopback mode ensures link quality between the device and a remote peer during installation or troubleshooting. In this mode, when the interface receives a frame that is not an OAM PDU or a pause frame, it sends it back on the same interface on which it was received. The link appears to be in the active state. You can use the returned loopback acknowledgement to test delay, jitter, and throughput.

  Junos OS can place a remote data terminal equipment (DTE) into loopback mode (if remote loopback mode is supported by the remote DTE). When you place a remote DTE into loopback mode, the interface receives the remote loopback request and puts the interface into remote loopback mode. When the interface is in remote loopback mode, all frames except OAM PDUs are looped back without any changes made to the frames. OAM PDUs continue to be sent and processed.

Table 117 on page 810 lists the interfaces modes supported.

Table 117: Supported Interface Modes

Interfaces	Mode
Physical interface (fe/ge)	Family • ccc • ethernet-switching • inet6 • inet • iso • mpls • tcc
	IFD encapsulations • ethernet-ccc • extended-vlan-ccc (IFD vlan-tagging mode) • ethernet-tcc • extended-vlan-tcc

Table 117: Supported Interface Modes (continued)

Interfaces	Mode
Aggregated Ethernet interface (Static or LACP lag)	Family • ethernet-switching • inet • mpls • iso • inet6
	IFD encapsulations • ethernet-ccc • extended-vlan-ccc (IFD vlan-tagging mode) • vlan-ccc

Related Documentation

- Example: Configuring Ethernet OAM Link Fault Management on a Security Device on page 811

Example: Configuring Ethernet OAM Link Fault Management on a Security Device

Starting in Junos OS Release 15.1X49-D70, configuring Ethernet OAM link fault management is supported on SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX1500 devices.

The Ethernet interfaces on the SRX Series devices support the IEEE 802.3ah standard for Operation, Administration, and Maintenance (OAM). The standard defines OAM link fault management (LFM). You can configure IEEE 802.3ah OAM LFM on point-to-point Ethernet links that are connected either directly or through Ethernet repeaters.

This example describes how to enable and configure OAM LFM on a Gigabit Ethernet or Fast Ethernet interface:

Requirements

This example uses the following hardware and software components:

- Junos OS Release 12.1 R2 or later for SRX Series Services Gateways
- Any two models of SRX Series devices connected directly

Before you begin:

- Establish basic connectivity. See the Getting Started Guide for your device.

- Configure network interfaces as necessary. See *Example: Creating an Ethernet Interface*.
- Ensure that you configure the interfaces as per the interface modules listed in "Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways" on page 809

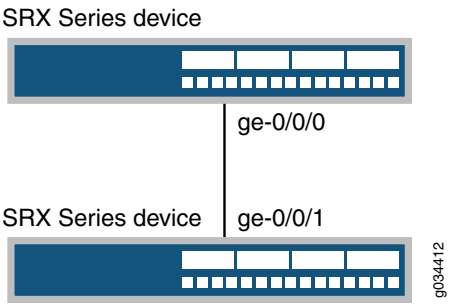
Overview

The Ethernet interfaces on the SRX Series devices support the IEEE 802.3ah standard for Operation, Administration, and Maintenance (OAM). The standard defines OAM link fault management (LFM). You can configure IEEE 802.3ah OAM LFM on point-to-point Ethernet links that are connected either directly or through Ethernet repeaters.

This example uses two SRX Series devices connected directly. Before you begin configuring Ethernet OAM LFM on these two devices, connect the two devices directly through supported interfaces. See "Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways" on page 809.

Figure 50 on page 812 shows the topology used in this example.

Figure 50: Ethernet LFM with SRX Series Devices



NOTE: For more information about configuring Ethernet OAM Link Fault Management, see Junos® OS Ethernet Interfaces.

Configuration

To configure Ethernet OAM LFM, perform these tasks:

- Configuring Ethernet OAM Link Fault Management on Device 1 on page 812
- Configuring Ethernet OAM Link Fault Management on Device 2 on page 813

Configuring Ethernet OAM Link Fault Management on Device 1

CLI Quick Configuration

To quickly configure this example, copy the following command, paste it into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the command into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols oam ethernet link-fault-management interface ge-0/0/0
```

```
set protocols oam ethernet link-fault-management interface ge-0/0/0 pdu-interval 800
set protocols oam ethernet link-fault-management interface ge-0/0/0 link-discovery
  active
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the Junos OS CLI User Guide.

To configure Ethernet OAM LFM on device 1:

1. Enable IEEE 802.3ah OAM support.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device1# set interface ge-0/0/0
   ```

2. Set the periodic OAM PDU-sending interval (in milliseconds) for fault detection.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device1# set interface pdu-interval 800
   ```

3. Specify that the interface initiates the discovery process.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device1# set interface ge-0/0/0 link-discovery active
   ```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@device1# show protocols
protocols {
  oam {
    ethernet {
      link-fault-management {
        interface ge-0/0/0 {
          pdu-interval 800;
          link-discovery active;
        }
      }
    }
  }
}
```

Configuring Ethernet OAM Link Fault Management on Device 2

CLI Quick Configuration

To quickly configure this example, copy the following command, paste it into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the command into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols oam ethernet link-fault-management interface ge-0/0/1
```

```
set protocols oam ethernet link-fault-management interface ge-0/0/1 pdu-interval 800
set protocols oam ethernet link-fault-management interface ge-0/0/1 negotiation-options
  allow-remote-loopback
```

Step-by-Step Procedure

To configure Ethernet OAM LFM on device 2:

1. Enable OAM on the peer interface.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device2# set interface ge-0/0/1
   ```

2. Set the periodic OAM PDU-sending interval (in milliseconds) for fault detection.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device2# set interface ge-0/0/1 pdu-interval 800
   ```

3. Enable remote loopback support for the local interface.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device2# set interface ge-0/0/1 negotiation-options allow-remote-loopback
   ```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@device2# show protocols
protocols {
  oam {
    ethernet {
      link-fault-management {
        interface ge-0/0/1 {
          negotiation-options {
            allow-remote-loopback;
          }
        }
      }
    }
  }
}
```

Verification

Verify the OAM LFM Configuration

Purpose

Verify that OAM LFM is configured properly.

Action

From operational mode, enter the **show oam ethernet link-fault-management** command.

```
user@device1> show oam ethernet link-fault-management

Interface: ge-0/0/0.0
Status: Running, Discovery state: Send Any
Peer address: 2001:bd8:00:31
Flags:Remote-Stable Remote-State-Valid Local-Stable 0x50
Remote entity information:
Remote MUX action: forwarding, Remote parser action: forwarding
Discovery mode: active, Unidirectional mode: unsupported
Remote loopback mode: supported, Link events: supported
Variable requests: unsupported
```

Meaning The output displays the MAC address and the discovery state is **Send Any** if OAM LFM has been configured properly.

Related Documentation

- Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways on page 809

Example: Configuring Remote Loopback Mode on VDSL Interfaces on a Security Device

Starting in Junos OS Release 15.1X49-D110, configuring remote loopback mode in Ethernet OAM link fault management (LFM) on a VDSL interface is supported on SRX320, SRX340, SRX345, and SRX550M devices.

This example describes the following configuration scenarios:

Starting in Junos OS Release 12.3X48-D65, configuring remote loopback mode in Ethernet OAM link fault management (LFM) on a VDSL interface is supported on SRX210, SRX220, SRX240, and SRX550 devices.

This example describes the following configuration scenarios:

- Scenario 1: Configuring remote loopback mode on a VDSL interface.
- Scenario 2: Configuring remote loopback mode on a VDSL interface acting as a PPPOE's underlying interface.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 15.1X49-D110 or later for SRX Series Services Gateways
- An SRX 210/220/240/320/340/345/550/550M device connected with a DSLAM

Before you begin:

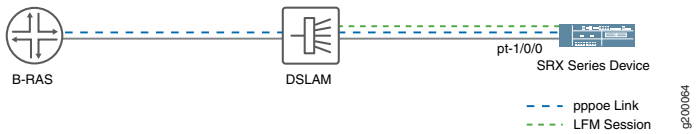
- Establish basic connectivity. See the Getting Started Guide for your device.
- Configure network interfaces as necessary. See *Example: Configuring VDSL2 Interfaces (Basic)*.
- Ensure that you configure the interfaces as per the interface modules listed in "Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways" on page 809
- Ensure that you configure PPPOE as per the instructions listed in *Example: Configuring PPPoE Interfaces*

Overview

This example uses an SRX Series device connected to a DSLAM. Before you begin configuring Ethernet OAM LFM on these two devices, connect the two devices directly through supported interfaces. See "Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways" on page 809.

Figure 50 on page 812 shows the topology used in this example.

Figure 51: Ethernet LFM with SRX Series Devices



NOTE: For more information about configuring Ethernet OAM Link Fault Management, see Junos® OS Ethernet Interfaces.

Configuration for Scenario 1

To configure remote loopback mode on a VDSL interface, perform these tasks:

- Configuring Remote Loopback Mode on a VDSL interface of an SRX Series Device on page 816

Configuring Remote Loopback Mode on a VDSL interface of an SRX Series Device

CLI Quick Configuration

To quickly configure this example, copy the following command, paste it into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the command into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols oam ethernet link-fault-management interface pt-1/0/0
set protocols oam ethernet link-fault-management interface pt-1/0/0 negotiation-options
    allow-remote-loopback
```

Step-by-Step Procedure

To configure remote loopback mode on a VDSL interface of an SRX Series device:

1. Enable OAM on a VDSL interface.

```
[edit protocols oam ethernet link-fault-management]
user@device2# set interface pt-1/0/0
```

2. Enable remote loopback support for the interface.

```
[edit protocols oam ethernet link-fault-management]
user@device2# set interface pt-1/0/0 negotiation-options allow-remote-loopback
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@device2# show protocols
protocols {
    oam {
        ethernet {
            link-fault-management {
                interface pt-1/0/0 {
                    negotiation-options {
                        allow-remote-loopback;
                    }
                }
            }
        }
    }
}
```

Configuration for Scenario 2

To configure remote loopback mode on a PPPOE's underlying interface, perform these tasks:

- Configuring Remote Loopback Mode on a PPPOE's underlying interface on page 817

Configuring Remote Loopback Mode on a PPPOE's underlying interface

CLI Quick Configuration

To quickly configure this example, copy the following command, paste it into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the command into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces pp0 unit 0 pppoe-options underlying-interface pt-1/0/0
set protocols oam ethernet link-fault-management interface pt-1/0/0 link-discovery
    active
set protocols oam ethernet link-fault-management interface pt-1/0/0 negotiation-options
    allow-remote-loopback
```

Step-by-Step Procedure

To configure remote loopback mode on a PPPOE's underlying interface:

1. Create the PPPoE interface pp0 and specify the logical PT interface pt-1/0/0 as the underlying interface.

   ```
   [edit protocols oam ethernet link-fault-management]
   user@device2# set interfaces pp0 unit 0 pppoe-options underlying-interface
     pt-1/0/0
   ```

2. Specify that the interface initiates the discovery process.

   ```
   user@device2# set protocols oam ethernet link-fault-management interface
     pt-1/0/0 link-discovery active
   ```

3. Enable remote loopback mode.

   ```
   user@device2# set protocols oam ethernet link-fault-management interface
     pt-1/0/0 negotiation-options allow-remote-loopback
   ```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@device2# show protocols
protocols {
  oam {
    ethernet {
      link-fault-management {
        interface pt-1/0/0 {
          link-discovery active;
          negotiation-options {
            allow-remote-loopback;
          }
        }
      }
    }
  }
}
```

Verification

Verify the OAM LFM Configuration

Purpose

Verify that OAM LFM is configured properly.

Action

From operational mode, enter the **show oam ethernet link-fault-management** command.

```
user@device1> show oam ethernet link-fault-management

Interface: pt-1/0/0.0
Status: Running, Discovery state: Send Any
```

```
Transmit interval: 300ms, PDU threshold: 3 frames, Hold time: 900ms
Peer address: 2001:db8:e5:b9:c8:ed
Flags:Remote-Stable Remote-State-Valid Local-Stable 0x50
Loopback tracking: Disabled, Loop status: Unknown
Remote entity information:
Remote MUX action: forwarding, Remote parser action: forwarding
Discovery mode: active, Unidirectional mode: unsupported
Remote loopback mode: unsupported, Link events: supported
Variable requests: unsupported
```

Meaning The output displays the MAC address and the discovery state is **Send Any** if OAM LFM has been configured properly.

Related Documentation

- Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways on page 809

CHAPTER 37

Configuring Ethernet OAM Link Fault Management on Security Devices

Understanding Ethernet OAM Connectivity Fault Management

Ethernet interfaces on SRX Series devices support the IEEE 802.1ag standard for Operation, Administration, and Management (OAM). The 802.1ag is an IEEE standard for connectivity fault management (CFM). The IEEE 802.1ag provides a specification for Ethernet CFM. The Ethernet network can consist of one or more service instances. A service instance could be a VLAN or a concatenation of VLANs. The goal of CFM is to provide a mechanism to monitor, locate, and isolate faulty links.

NOTE: **Support for the IEEE 802.1ag standard for OAM on SRX Series devices depends on the Junos OS release running on the device.**

Starting in Junos OS Release 15.1X49-D80, Ethernet OAM CFM is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, Ethernet OAM CFM is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Ethernet OAM CFM is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

CFM support includes the following features:

- Fault monitoring using the Continuity Check Protocol. This is a neighbor discovery and health check protocol that discovers and maintains adjacencies at the VLAN or link level.
- Path discovery and fault verification using the Link Trace protocol. This feature is not supported in Junos OS Release 12.3X48-D65.
- Fault isolation using the Loopback protocol.

  The Loopback protocol is used to check access to maintenance association end points (MEPs) under the same maintenance association (MA). The Loopback messages are triggered by an administrator using the **ping ethernet** command.

NOTE: Virtual private LAN service (VPLS) is not supported on SRX100, SRX110, SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX550M, SRX1400, and SRX1500 devices.

CFM partitions the service network into various administrative domains. For example, operators, providers, and customers might be part of different administrative domains. Each administrative domain is mapped into one maintenance domain providing enough information to perform its own management, thus avoiding security breaches and making end-to-end monitoring possible.

In a CFM maintenance domain, each service instance is called a maintenance association. A maintenance association can be thought of as a full mesh of maintenance association end points (MEPs) having similar characteristics. MEPs are active CFM entities generating and responding to CFM protocol messages. There is also a maintenance association intermediate point (MIP), which is a CFM entity similar to the MEP, but more passive (MIPs only respond to CFM messages).

Each maintenance domain is associated with a maintenance domain level from 0 through 7. Level allocation is based on the network hierarchy, where outer domains are assigned a higher level than the inner domains. You configure customer end points to have the highest maintenance domain level. The maintenance domain level is a mandatory parameter that indicates the nesting relationships between various maintenance domains. The level is embedded in each CFM frame. CFM messages within a given level are processed by MEPs at that same level.

To enable CFM on an Ethernet interface, you must configure maintenance domains, maintenance associations, and MEPs.

The limitations for CFM are as follows:

- You cannot configure MEP and MIP on the same VLAN.
- CFM and link fault management (LFM) can be configured on the same interface.
- You cannot configure CFM with Generic VLAN Registration Protocol (GVRP).

- CFM is not supported on VoIP VLAN ports.
- On SRX240, and SRX550M devices, the default Loopback message (LBM) packet size is 113 bytes.

Benefits of Ethernet CFM

Ethernet CFM provides the following benefits:

- End-to-end service-level OAM technology
- Reduced operating expense for service provider Ethernet networks
- Competitive advantage for service providers

CFM over VDSL and PPPoE interfaces for SRX210, SRX220, SRX240, SRX320, SRX340, SRX345, SRX550, and SRX550M Devices

Starting in Junos OS Release 12.3X48-D65, on SRX210, SRX220, SRX240, and SRX550 devices, Operation, Administration, and Maintenance (OAM) connectivity fault management (CFM) is supported on very-high-bit-rate digital subscriber line (VDSL) and Point-to-Point Protocol over Ethernet (PPPoE) interfaces in addition to Ethernet interfaces.

CFM over VDSL should be configured on the pt interface. To support CFM over PPPoE, you need to configure maintenance domain and maintenance association end point (MEP). The CFM over VDSL interface supports down direction MEP, continuity check, and loopback protocols.

The following are the limitations when configuring Ethernet CFM over VDSL or Layer 3 interface:

- CFM action profiles are not supported on the Point-to-Point Protocol over Ethernet (PPPoE) logical interface on SRX210, SRX220, SRX240, SRX550, and SRX650 devices.
- Synthetic loss measurement on demand is supported only on SRX320, SRX340, SRX345, and SRX550M devices. Proactive synthetic loss measurement is not supported.
- When CFM over PPPoE is implemented, CFM must be applied on the PPPoE logical interface and not on the underlying interface.
- CFM over VDSL can be implemented as a MEP but not as a MIP.
- CFM higher-level pass-through over a VDSL or Gigabit Ethernet interface in Layer 3 interface mode is not supported.
- For a VLAN-tagged VDSL interface, CFM must always be applied on the respective logical interface and not over the physical interface.
- When CFM is enabled on VDSL, CFM packets are dropped randomly, causing CFM sessions to flap based on the timer when transit traffic exceeds the line rate. Flapping occurs because the VDSL Mini-Physical Interface Module (Mini-PIM) cannot differentiate and prioritize CFM packets.

Related Documentation

- Example: Configuring Ethernet OAM Connectivity Fault Management on a Security Device on page 824

Example: Configuring Ethernet OAM Connectivity Fault Management on a Security Device

Starting in Junos OS Release 15.1X49-D80, Ethernet OAM connectivity fault management is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, Ethernet OAM connectivity fault management is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Ethernet OAM connectivity fault management is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

Connectivity Fault Management (CFM) provides a mechanism to monitor, locate, and isolate faulty links.

This example describes how to enable and configure an end-to-end OAM CFM session on an Ethernet interface.

Requirements

This example uses the following hardware and software components:

- Three SRX Series devices connected by a point-to-point Ethernet link.
- Junos OS Release 12.1X44-D10 or later for SRX Series devices.

Overview

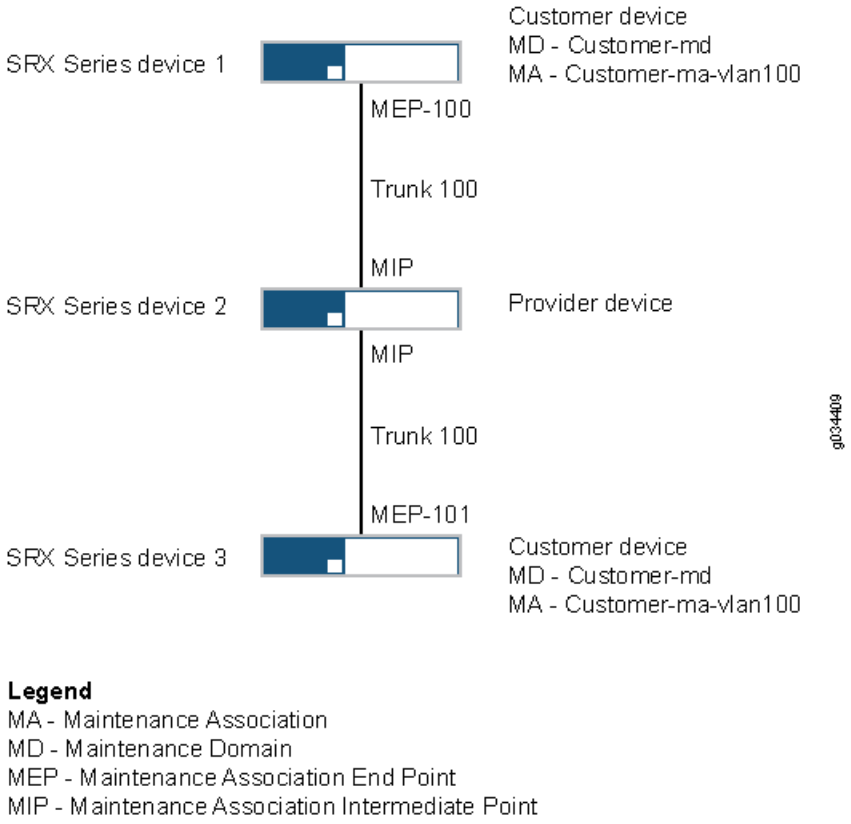
Ethernet interfaces on SRX Series devices support the IEEE 802.1ag standard for Operation, Administration, and Management (OAM). The IEEE 802.1ag specification provides a specification for Ethernet connectivity fault management (CFM). CFM can be used to detect faults in the network path between the customer premises devices. It also helps in detecting the device or node in the provider network, where the failure occurred.

This example describes how to configure an end to end CFM session. In this example, three devices are connected by a point-to-point Ethernet link. The link between these devices is monitored using CFM. To check connectivity or fault through the provider network, maintenance intermediate point (MIP) is configured.

Topology

Figure 52 on page 825 shows three SRX Series devices connected by a point-to-point Ethernet link.

Figure 52: Ethernet CFM with SRX Series Devices



Legend
MA - Maintenance Association
MD - Maintenance Domain
MEP - Maintenance Association End Point
MIP - Maintenance Association Intermediate Point

Configuring Ethernet OAM Connectivity Fault Management

Configuring Ethernet OAM Connectivity Fault Management on Device 1

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members v100
set vlans v100 vlan-id 100
set protocols oam ethernet connectivity-fault-management maintenance-domain
    Customer-md level 5
set protocols oam ethernet connectivity-fault-management maintenance-domain
    Customer-md maintenance-association Customer-ma mep 100 interface ge-0/0/4.0
set protocols oam ethernet connectivity-fault-management maintenance-domain
    Customer-md maintenance-association Customer-ma mep 100 interface vlan 100
```

```
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma mep 100 auto-discovery
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma continuity-check interval 10s
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma continuity-check hold-interval
  20
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To enable and configure OAM CFM on device 1:

1. Define a VLAN and enable the interface for family Ethernet switching with interface mode trunk or access.

```
[edit]
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode
  trunk
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members
  v100
user@host# set vlans v100 vlan-id 100
```

2. Specify the maintenance domain name and the maintenance domain level.

```
[edit protocols oam ethernet connectivity-fault-management ]
user@host# set maintenance-domain Customer-md level 5
```

3. Create a maintenance association and configure MEP.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md]
user@host# set maintenance-association Customer-ma mep 100 interface
  ge-0/0/4.0
user@host# set maintenance-association Customer-ma mep 100 interface vlan
  100
```

4. Enable MEP automatic discovery.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma]
user@host# set mep 100 auto-discovery
```

5. Enable the Continuity Check Protocol and specify the continuity check interval and hold interval.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma]
user@host# set continuity-check interval 10s
user@host# set continuity-check hold-interval 20
```

Results From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show protocols** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show protocols

oam {
  ethernet {
    connectivity-fault-management {
      maintenance-domain Customer-md {
        level 5;
        maintenance-association Customer-ma {
          continuity-check {
            interval 10s;
            hold-interval 20;
          }
          mep 100 {
            interface ge-0/0/4.0 vlan 100;
            auto-discovery;
          }
        }
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Ethernet OAM CFM with MIP Half Function on Device 2

CLI Quick Configuration To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members v100
set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members v100
set vlans v100 vlan-id 100
set protocols oam ethernet connectivity-fault-management maintenance-domain
  default-5 v100
set protocols oam ethernet connectivity-fault-management maintenance-domain
  default-5 mip-half-function default
```


Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure MIP half function:

1. Define a VLAN and enable the interface for family Ethernet switching with interface mode trunk or access.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode
  trunk
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
  v100
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching interface-mode
  trunk
user@host# set interfaces ge-0/0/4 unit 0 family ethernet-switching vlan members
  v100
user@host# set vlans v100 vlan-id 100
```

2. Create a maintenance domain and configure VLAN.

```
[edit protocols oam ethernet connectivity-fault-management]
user@host# set maintenance-domain default-5 v100
```

3. Create a MIP half function.

```
[edit protocols oam ethernet connectivity-fault-management ]
user@host# set maintenance-domain default-5 mip-half-function default
```

NOTE: If you want to configure traceoptions, run the following commands:

```
set protocols oam ethernet connectivity-fault-management traceoptions
  file CFM_trace
set protocols oam ethernet connectivity-fault-management traceoptions
  flag all
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show protocols
  oam {
    ethernet {
```

```
        connectivity-fault-management {
          traceoptions {
            file CFM_trace;
            flag all;
          }
          maintenance-domain default-5 {
            v100;
            mip-half-function default;
          }
        }
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring Ethernet OAM Connectivity Fault Management on Device 3

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members v100
set vlans v100 vlan-id 100
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md level 5
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma mep 101 interface ge-0/0/1.0
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma mep 101 interface vlan 100
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma mep 101 auto-discovery
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma continuity-check hold-interval
  20
set protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma continuity-check interval 10s
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To enable and configure OAM CFM on Device 3:

1. Define a VLAN and enable the interface for family Ethernet switching with interface mode trunk or access.

```
[edit]
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode
  trunk
user@host# set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members
  v100
user@host# set vlans v100 vlan-id 100
```

2. Specify the maintenance domain name and the maintenance domain level.

```
[edit protocols oam ethernet connectivity-fault-management ]
user@host# set maintenance-domain Customer-md level 5
```

3. Create a maintenance association and configure MEP.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md]
user@host# set maintenance-association Customer-ma mep 101 interface
  ge-0/0/1.0
user@host# set maintenance-association Customer-ma mep 101 interface vlan 100
```

4. Enable MEP automatic discovery.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md]
user@host# set maintenance-association Customer-ma mep 101 auto-discovery
```

5. Enable the Continuity Check Protocol and specify the continuity check interval and hold interval.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  Customer-md maintenance-association Customer-ma]
user@host# set continuity-check interval 10s
user@host# set continuity-check hold-interval 20
```

Results From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

For brevity, this **show** command output includes only the configuration that is relevant to this example. Any other configuration on the system has been replaced with ellipses (...).

```
[edit]
user@host# show protocols
  oam {
    ethernet {
      connectivity-fault-management {
        maintenance-domain Customer-md {
          level 5;
          maintenance-association Customer-ma {
            continuity-check {
              interval 10s;
              hold-interval 20;
            }
            mep 101 {
              interface ge-0/0/1.0 vlan 100;
              auto-discovery;
            }
          }
        }
```



```
        }
      }
    }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying the OAM CFM Configuration on Device 1

Purpose Verify that OAM CFM has been configured properly.

Action From operational mode, enter the following commands:

- **show oam ethernet connectivity-fault-management adjacencies** to display connectivity-fault-management adjacencies.
- **show oam ethernet connectivity-fault-management interfaces** to display the Ethernet OAM information for the specified interface.

These commands produce the following sample output:

```
user@host# show oam ethernet connectivity-fault-management adjacencies
Mep-id   Interface          State              Timer to Expire
          101     ge-0/0/4.0         ok                 29

user@host# show oam ethernet connectivity-fault-management interfaces
Interface         Link      Status       Level  MEP         Neighbours
                                                           Identifier
       ge-0/0/4.0          Up        Active       5      100         1

user@host# show oam ethernet connectivity-fault-management interfaces detail
Interface name: ge-0/0/4.0, vlan 100, Interface status: Active, Link status: Up
  Maintenance domain name: Customer-md, Format: string, Level: 5
    Maintenance association name: Customer-ma, Format: string
    Continuity-check status: enabled, Interval: 10s
    MEP identifier: 100, Direction: down, MAC address: 2c:6b:f5:62:29:84
    MEP status: running
    Defects:
      Remote MEP not receiving CCM                : no
      Erroneous CCM received                      : no
      Cross-connect CCM received                  : no
      RDI sent by some MEP                        : no
    Statistics:
      CCMs sent                                   : 7
      CCMs received out of sequence               : 0
```

```
    LBMs sent                                      : 0
    Valid in-order LBRs received                   : 0
    Valid out-of-order LBRs received               : 0
    LBRs received with corrupted data              : 0
    LBRs sent                                      : 0
    LTMs sent                                      : 0
    LTMs received                                  : 0
    LTRs sent                                      : 0
    LTRs received                                  : 0
    Sequence number of next LTM request            : 0
    1DMs sent                                      : 0
    Valid 1DMs received                            : 0
    Invalid 1DMs received                          : 0
    DMMs sent                                      : 0
    DMRs sent                                      : 0
    Valid DMRs received                            : 0
    Invalid DMRs received                          : 0
  Remote MEP count: 1
    Identifier    MAC address        State    Interface
      101     80:71:1f:ad:53:81       ok    ge-0/0/4.0
```

Meaning

- If the **show oam ethernet connectivity-fault-management interfaces detail** command output displays continuity-check status as **enabled** and displays details of the remote MEP, it means that connectivity fault management (CFM) was configured properly.
- If the **show oam ethernet connectivity-fault-management adjacencies** command output displays the state as **ok**, it indicates that the Continuity Check Protocol is up.

Verifying the OAM CFM Configuration with MIP Half Function on Device 2

Purpose Verify that OAM CFM has been configured properly.

Action From operational mode, run the **show oam ethernet connectivity-fault-management mip** command.

```
user@host# show oam ethernet connectivity-fault-management mip vlan 100
default maintenance-domain mhf    : default

  Interface       Level
  ge-0/0/1.0          5
  ge-0/0/4.0          5
```

Meaning The **show oam ethernet connectivity-fault-management mip** command output displays the MIP information.

Verifying the OAM CFM Configuration on Device 3

Purpose Verify that OAM CFM has been configured properly.

Action From operational mode, enter the following commands:

- **show oam ethernet connectivity-fault-management adjacencies** to display connectivity-fault-management adjacencies.
- **show oam ethernet connectivity-fault-management interfaces** to display the Ethernet OAM information for the specified interface.

```
user@host# show oam ethernet connectivity-fault-management adjacencies
Mep-id      Interface          State       Timer to Expire
          100       ge-0/0/1.0         ok            27
```

```
user@host# show oam ethernet connectivity-fault-management interfaces detail
Interface name: ge-0/0/1.0, vlan 100, Interface status: Active, Link status: Up
  Maintenance domain name: Customer-md, Format: string, Level: 5
    Maintenance association name: Customer-ma, Format: string
    Continuity-check status: enabled, Interval: 10s
    MEP identifier: 101, Direction: down, MAC address: 80:71:1f:ad:53:81
    MEP status: running
    Defects:
      Remote MEP not receiving CCM                 : no
      Erroneous CCM received                       : no
      Cross-connect CCM received                   : no
      RDI sent by some MEP                         : no
    Statistics:
      CCMs sent                                    : 77
      CCMs received out of sequence                : 0
      LBMs sent                                    : 0
      Valid in-order LBRs received                 : 0
      Valid out-of-order LBRs received             : 0
      LBRs received with corrupted data            : 0
      LBRs sent                                    : 0
      LTMs sent                                    : 0
      LTMs received                                : 0
      LTRs sent                                    : 0
      LTRs received                                : 0
      Sequence number of next LTM request          : 0
      1DMs sent                                    : 0
      Valid 1DMs received                          : 0
      Invalid 1DMs received                        : 0
      DMMs sent                                    : 0
      DMRs sent                                    : 0
      Valid DMRs received                          : 0
      Invalid DMRs received                        : 0
    Remote MEP count: 1
      Identifier    MAC address        State    Interface
        100     2c:6b:f5:62:29:84       ok    ge-0/0/1.0
```

Meaning

- If the **show oam ethernet connectivity-fault-management interfaces detail** command output displays continuity-check status as **enabled** and displays details of the remote MEP, it means that connectivity fault management (CFM) was configured properly.
- If the **show oam ethernet connectivity-fault-management adjacencies** command output displays the state as **ok**, it indicates that the Continuity Check Protocol is up.

Verifying the Path Using the Link Trace Protocol

Purpose Verify the path between maintenance endpoints.

Action From operational mode, enter the **traceroute ethernet** command.

```
user@host# traceroute ethernet maintenance-domain Customer-md maintenance-association
Customer-ma mep 101
Linktrace to 80:71:1f:ad:53:81, Interface : ge-0/0/4.0
      Maintenance Domain: Customer-md, Level: 5
      Maintenance Association: Customer-ma, Local Mep: 100
      Transaction Identifier: 3
    Hop   TTL    Source MAC address        Next-hop MAC address
    .
    1     63     80:71:1f:ad:50:01         80:71:1f:ad:50:01
    2     62     80:71:1f:ad:53:81         00:00:00:00:00:00
```

Verifying MEP Continuity Using Ping

Purpose Verify access to MEPs under the same maintenance association.

Action From operational mode, enter the **ping ethernet** command.

```
user@host# ping ethernet maintenance-domain Customer-md maintenance-association
Customer-ma mep 101
PING to 80:71:1f:ad:53:81, Interface ge-0/0/4.0
60 bytes from 80:71:1f:ad:53:81: lbm_seq=0
60 bytes from 80:71:1f:ad:53:81: lbm_seq=1
60 bytes from 80:71:1f:ad:53:81: lbm_seq=2
60 bytes from 80:71:1f:ad:53:81: lbm_seq=3
--- ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
```

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821

Creating a Maintenance Domain on a Security Device

A maintenance domain consists of network entities such as operators, providers, and customers. A maintenance domain is a management space for managing and administering a network. A domain is owned and operated by a single entity and defined by the set of ports internal to it and at its boundary. Each maintenance domain is associated with a maintenance domain level from 0 through 7. Level allocation is based on the network hierarchy, where outer domains are assigned a higher level than the inner domains. You configure customer end points to have the highest maintenance domain level. The maintenance domain level is a mandatory parameter that indicates the nesting relationships between various maintenance domains.

To enable connectivity fault management (CFM) on an Ethernet interface, maintenance domains, maintenance associations, and maintenance association end points (MEPs) must be created and configured.

Starting in Junos OS Release 12.3X48-D65, on SRX210, SRX220, SRX240, and SRX550 devices, creating a maintenance domain for Ethernet OAM CFM is supported over VDSL and PPPoE interfaces in addition to Ethernet interfaces.

Starting in Junos OS Release 15.1X49-D80, creating a maintenance domain for Ethernet OAM CFM is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, creating a maintenance domain for Ethernet OAM CFM is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Creating a maintenance domain for Ethernet OAM CFM is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

To create a maintenance domain:

1. Specify a name for the maintenance domain.

   [edit protocols oam ethernet connectivity-fault-management]
   user@host# set maintenance-domain *domain-name*

2. Specify a format for the maintenance domain name. If you do not specify a format, no name is configured.

   - A plain ASCII character string
   - A Domain Name System (DNS) format
   - A media access control (MAC) address plus a two-octet identifier in the range 0 through 65,535
   - None

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name*]
   user@host# set name-format *format*

   For example, to specify the name format as a MAC address plus a two-octet identifier:

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name*]

   user@host# set name-format *mac+2oct*

3. Configure the maintenance domain level, which is used to indicate the nesting relationship between this domain and other domains. Use a value from 0 through 7.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name*]

   user@host# set level *level-number*

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821
- Configuring the Continuity Check Protocol on a Security Device on page 839
- Configuring a Maintenance Domain MIP Half Function on a Security Device on page 838
- Creating a Maintenance Association on a Security Device on page 836
- Configuring a Maintenance Association End Point on a Security Device on page 837
- Configuring the Link Trace Protocol on a Security Device on page 841

Creating a Maintenance Association on a Security Device

In a connectivity fault management (CFM) maintenance domain, each service instance is called a maintenance association. A maintenance association can be thought of as a full mesh of maintenance association end points (MEPs) having similar characteristics.

Starting in Junos OS Release 12.3X48-D65, on SRX210, SRX220, SRX240, and SRX550 devices, creating a maintenance association for Ethernet OAM connectivity fault management is supported over VDSL and PPPoE interfaces in addition to Ethernet interfaces.

Starting in Junos OS Release 15.1X49-D80, creating a maintenance association for Ethernet OAM CFM is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, creating a maintenance association for Ethernet OAM CFM is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Creating a maintenance association for Ethernet OAM CFM is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

To create a maintenance association:

[edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name*]
user@host# set maintenance-association *ma-name*

NOTE: On SRX210, SRX220, SRX240, SRX300, SRX320, SRX340, SRX345, SRX550, SRX550M, and SRX650 devices, a maximum of seven maintenance associations are supported.

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821
- Creating a Maintenance Domain on a Security Device on page 834
- Configuring a Maintenance Domain MIP Half Function on a Security Device on page 838
- Configuring the Continuity Check Protocol on a Security Device on page 839
- Configuring a Maintenance Association End Point on a Security Device on page 837
- Configuring the Link Trace Protocol on a Security Device on page 841

Configuring a Maintenance Association End Point on a Security Device

Starting in Junos OS Release 12.3X48-D65, on SRX210, SRX220, SRX240, and SRX550 devices, configuring a maintenance association end point for Ethernet OAM CFM is supported over VDSL and PPPoE interfaces in addition to Ethernet interfaces.

Starting in Junos OS Release 15.1X49-D80, configuring a maintenance association end point for Ethernet OAM CFM is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, configuring a maintenance association end point for Ethernet OAM CFM is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Configuring a maintenance association end point for Ethernet OAM CFM is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

To configure a MEP:

1. Specify an ID for the MEP. The value can be from 1 through 8191.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name*]
   user@host# set mep *mep-id*

2. Enable MEP automatic discovery if you want to have the MEP accept continuity check messages (CCMs) from all remote MEPs of the same maintenance association.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name* mep *mep-id*]
   user@host# set auto-discovery

3. Specify that CFM CCM packets be transmitted only in one direction for the MEP. That is, set the direction as down so that CCMs are transmitted only out of (not into) the interface configured on this MEP.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name* mep *mep-id*]
   user@host# set direction down

4. Specify the logical interface to which the MEP is attached. It can be either an access interface or a trunk interface. If you specify a trunk interface, the VLAN associated with that interface must have a VLAN ID.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name* mep *mep-id*]
   user@host# set interface *interface-name*

5. Configure a remote MEP from which CCMs are expected. If automatic discovery is not enabled, the remote MEP must be configured under the **mep** statement; otherwise, the CCMs from the remote MEP will be treated as errors.

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  domain-name maintenance-association ma-name mep mep-id]
user@host# set remote-mep mep-id
```

NOTE: You cannot configure MEPs at different levels for the same VLANs.

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821
- Creating a Maintenance Domain on a Security Device on page 834
- Configuring a Maintenance Domain MIP Half Function on a Security Device on page 838
- Creating a Maintenance Association on a Security Device on page 836
- Configuring the Continuity Check Protocol on a Security Device on page 839
- Configuring the Link Trace Protocol on a Security Device on page 841

Configuring a Maintenance Domain MIP Half Function on a Security Device

Starting in Junos OS Release 15.1X49-D80, configuring a maintenance domain MIP half function for Ethernet OAM connectivity fault management is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, configuring a maintenance domain MIP half function for Ethernet OAM connectivity fault management is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Configuring a maintenance domain MIP half function for Ethernet OAM connectivity fault management is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

MIP half function (MHF) divides the maintenance association intermediate point (MIP) functionality into two unidirectional segments, improves visibility with minimal configuration, and improves network coverage by increasing the number of points that can be monitored. MHF extends monitoring capability by responding to loopback and Link Trace messages to help isolate faults. Whenever a MIP is configured, the MIP half function value for all maintenance domains and maintenance associations must be the same.

To configure the MIP half function:

```
[edit protocols oam ethernet connectivity-fault-management maintenance-domain
  domain-name]
user@host# set mip-half-function default
```

NOTE:

- If SRX340, or SRX345 devices are configured as MIPs, ensure that a static MAC is configured in the Ethernet switching table with the next-hop interface to the MEP MAC.
- You cannot configure MIP in a nondefault domain.
- In Q-in-Q mode, double tag packets are not retained by MIP.
- A maximum of 116 MIPs can be configured on a device.

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821
- Creating a Maintenance Domain on a Security Device on page 834
- Creating a Maintenance Association on a Security Device on page 836
- Configuring the Continuity Check Protocol on a Security Device on page 839
- Configuring a Maintenance Association End Point on a Security Device on page 837
- Configuring the Link Trace Protocol on a Security Device on page 841

Configuring the Continuity Check Protocol on a Security Device

The Continuity Check Protocol is used for fault detection by a maintenance association end point (MEP) within a maintenance association. The MEP periodically sends continuity check multicast messages. The receiving MEPs use the continuity check messages (CCMs) to build a MEP database of all MEPs in the maintenance association.

Starting in Junos OS Release 12.3X48-D65, on SRX210, SRX220, SRX240, and SRX550 devices, the continuity check protocol for Ethernet Operation, Administration, and Management (OAM) connectivity fault management is supported over VDSL and PPPoE interfaces in addition to Ethernet interfaces.

Starting in Junos OS Release 15.1X49-D80, the continuity check protocol for Ethernet OAM CFM is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, the continuity check protocol for Ethernet OAM CFMis supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

The continuity check protocol for Ethernet OAM CFM is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

To configure the Continuity Check Protocol:

1. Enable the Continuity Check Protocol.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name*]
   user@host# set continuity-check

2. Specify the continuity check hold interval. The hold interval is the number of minutes to wait before flushing the MEP database if no updates occur. The default value is 10 minutes (not supported in Junos OS Release 12.3X48-D60).

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name* continuity-check]
   user@host# set hold-interval *number*

3. Specify the CCM interval. The interval is the time between the transmission of CCMs. You can specify 10 minutes (10m), 1 minute (1m), 10 seconds (10s), 1 second (1s), or 100 milliseconds (100ms).

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name* continuity-check]
   user@host# set interval *number*

4. Specify the number of CCMs (that is, protocol data units) that can be lost before the MEP is marked as down. The default number of protocol data units (PDUs) is 3.

   [edit protocols oam ethernet connectivity-fault-management maintenance-domain *domain-name* maintenance-association *ma-name* continuity-check]
   user@host# set loss-threshold *number*

NOTE: If the CCM interval is 100 milliseconds, only four MEPs are supported on a device.

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821
- Creating a Maintenance Domain on a Security Device on page 834
- Creating a Maintenance Association on a Security Device on page 836
- Configuring a Maintenance Domain MIP Half Function on a Security Device on page 838
- Configuring the Link Trace Protocol on a Security Device on page 841

Configuring the Link Trace Protocol on a Security Device

Starting in Junos OS Release 15.1X49-D80, configuring the Link Trace protocol for Ethernet OAM connectivity fault management is supported on SRX1500 devices.

Starting in Junos OS Release 15.1X49-D75, configuring the Link Trace protocol for Ethernet OAM connectivity fault management is supported on SRX300, SRX320, SRX340, SRX345, and SRX550M devices.

Configuring the Link Trace protocol for Ethernet OAM connectivity fault management is not supported from Junos OS Release 15.1X49-D40 to Junos OS Release 15.1X49-D70.

The Link Trace protocol is used for path discovery between a pair of maintenance points. Link Trace Messages (LTMs) are triggered by an administrator using the **traceroute ethernet** command to verify the path between a pair of MEPs under the same maintenance association. LTMs can also be used to verify the path between a MEP and a MIP under the same maintenance domain.

To configure the Link Trace protocol:

1. Configure the Link Trace path age timer. If no response to a Link Trace request is received, the request and response entries are deleted after the age timer expires.

   [edit protocols oam ethernet connectivity-fault-management]
   user@host# set linktrace age *time*

2. Configure the number of Link Trace Reply (LTR) entries to be stored per Link Trace request.

   [edit protocols oam ethernet connectivity-fault-management]
   user@host# set linktrace path-database-size *path-database-size*

Related Documentation

- Understanding Ethernet OAM Connectivity Fault Management on page 821
- Creating a Maintenance Domain on a Security Device on page 834
- Creating a Maintenance Association on a Security Device on page 836
- Configuring a Maintenance Domain MIP Half Function on a Security Device on page 838
- Configuring the Continuity Check Protocol on a Security Device on page 839

CHAPTER 38

Configuring Reflective Relay on Switches

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

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 and VLANs on Switches on page 84

Configuring Reflective Relay on Switches

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 on Switches with ELS Support" on page 845.

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 QFX Switches on page 846
- Understanding Reflective Relay for Use with VEPA Technology on page 843

Configuring Reflective Relay on Switches with ELS Support

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 supports the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring Reflective Relay on Switches" on page 844.

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 an interface mode of **trunk**:

   ```
   [edit]
   user@switch# set interfaces interface-name unit number family family-type interface-mode trunk
   ```

   For example:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode trunk
   ```

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 QFX Switches with ELS Support on page 851
- Understanding Reflective Relay for Use with VEPA Technology on page 843

Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches

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 on QFX Switches with ELS Support" on page 851.

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

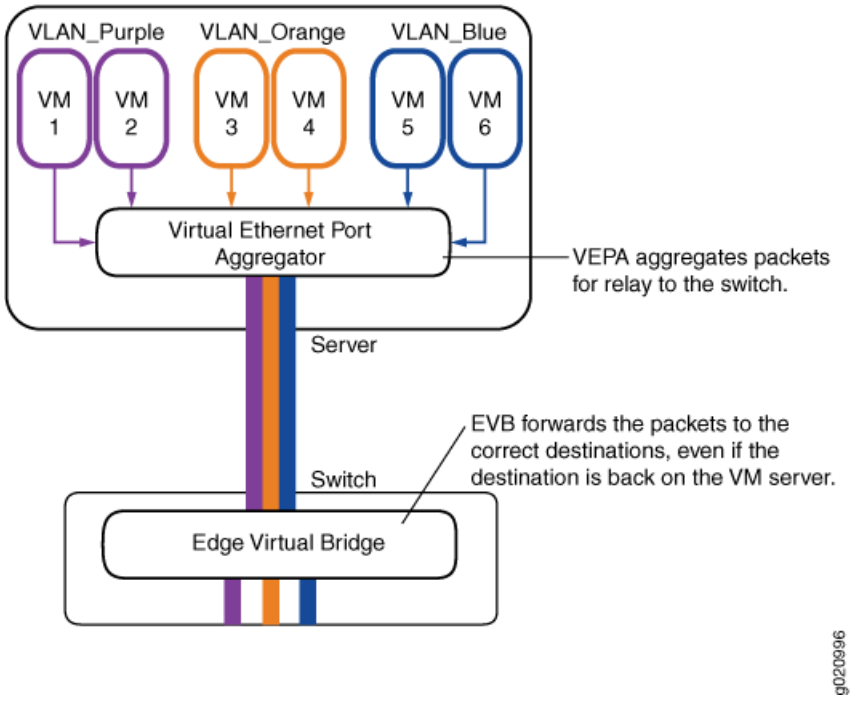
Before you configure reflective relay on a switch port, be sure you have:

- 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 53 on page 848, 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 53 on page 848 shows the topology for this example.

Figure 53: 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 118 on page 848 shows the components used in this example.

Table 118: Components of the Topology for Configuring Reflective Relay

Component	Description
QFX3500 switch	Switch that supports reflective relay.
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

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 850

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 843
- Configuring Reflective Relay on Switches on page 844
- Configuring VLANs on Switches on page 93

- port-mode on page 1072
- reflective-relay on page 1094

Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches with ELS Support

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 with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Example: Configuring Reflective Relay for Use with VEPA Technology on QFX Switches" on page 846. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

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

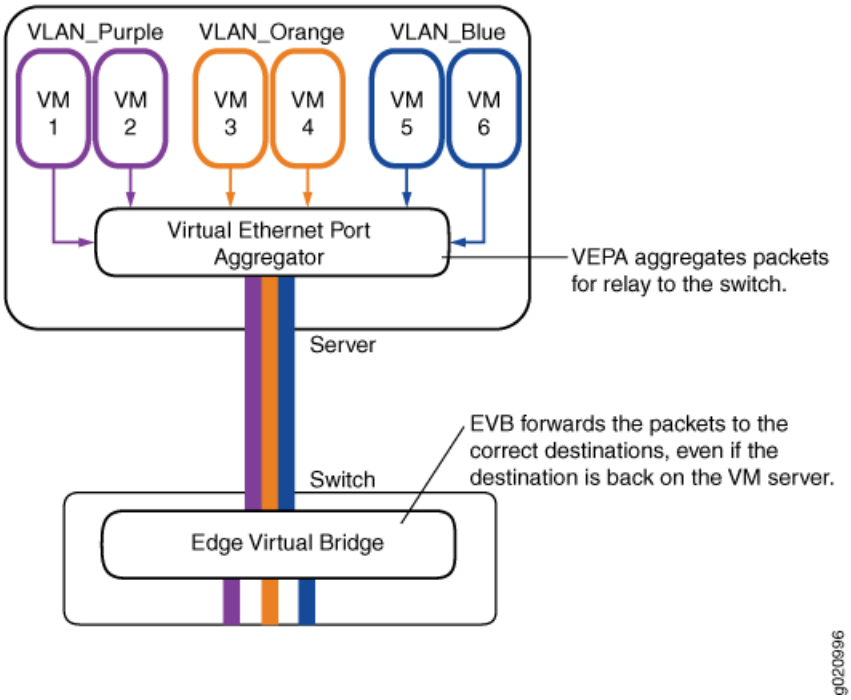
Before you configure reflective relay on a switch port, be sure you have:

- 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 53 on page 848, 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 53 on page 848 shows the topology for this example.

Figure 54: Reflective Relay Topology



In this example, you configure the physical Ethernet switch port interface for trunk interface mode and reflective relay. Configuring trunk 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 118 on page 848 shows the components used in this example.

Table 119: Components of the Topology for Configuring Reflective Relay

Component	Description
QFX3500 switch	Switch that supports reflective relay. .
xe-0/0/2	Switch interface to the server.

Table 119: Components of the Topology for Configuring Reflective Relay (continued)

Component	Description
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 853

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 interface-mode trunk
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 trunk interface mode on the interface:

   ```
   [edit]
   user@switch# set interfaces xe-0/0/2 unit 0 family ethernet-switching interface-mode
   trunk
   ```

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 {
        interface-mode trunk;
        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 854

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, Interface mode: Trunk
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 trunk interface 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 843
- *Configuring Port Mirroring*
- interface-mode on page 997
- reflective-relay on page 1094

CHAPTER 39

Configuring Edge Virtual Bridging

Understanding Edge Virtual Bridging for Use with VEPA Technology on EX Series Switches

Servers using virtual Ethernet port aggregator (VEPA) do not send packets directly from one virtual machine (VM) to another. Instead, the packets are sent to virtual bridges on an adjacent switch for processing. EX Series switches use edge virtual bridging (EVB) as a virtual bridge to return the packets on the same interface that delivered the packets.

What Is EVB?

EVB is a software capability on a switch running Junos OS that allows multiple virtual machines to communicate with each other and with external hosts in the Ethernet network environment.

What Is VEPA?

VEPA is a software capability on a server that collaborates with an adjacent, external switch to provide bridging support between multiple virtual machines and external networks. The VEPA collaborates with the adjacent switch by forwarding all VM-originated frames to the adjacent switch for frame processing and frame relay (including hairpin forwarding) and by steering and replicating frames received from the VEPA uplink to the appropriate destinations.

Why Use VEPA Instead of VEB?

Even though virtual machines are capable of sending packets directly to one another with a technology called virtual Ethernet bridging (VEB), you typically want to use physical switches for switching because VEB uses expensive server hardware to accomplish the task. Instead of using VEB, you can install VEPA on a server to offload switching functionality to an adjacent, less expensive physical switch. Additional advantages of using VEPA include:

- VEPA reduces complexity and allows higher performance at the server.
- VEPA takes advantage of the physical switch's security and tracking features.
- VEPA provides visibility of inter-virtual-machine traffic to network management tools designed for an adjacent bridge.
- VEPA reduces the amount of network configuration required by server administrators, and as a consequence, reduces work for the network administrator.

How Does EVB Work?

EVB uses two protocols, Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP) and Edge Control Protocol (ECP), to program policies for each individual virtual switch instance—specifically, EVB maintains the following information for each VSI instance:

- VLAN ID
- VSI type
- VSI type version
- MAC address of the server

VDP is used by the VEPA server to propagate VSI information to the switch. This allows the switch to program policies on individual VSIs and supports virtual machine migration by implementing logic to preassociate a VSI with a particular interface.

ECP is a Link Layer Discovery Protocol (LLDP)-like transport layer that allows multiple upper layer protocols to send and receive protocol data units (PDUs). ECP improves upon LLDP by implementing sequencing, retransmission and an ack mechanism, while at the same time remaining lightweight enough to be implemented on a single-hop network. ECP is implemented in an EVB configuration when you configure LLDP on interfaces that you have configured for EVB. That is, you configure LLDP, not ECP.

How Do I Implement EVB?

You can configure EVB on a switch when that switch is adjacent to a server that includes VEPA technology. In general, this is what you do to implement EVB:

- The network manager creates a set of VSI types. Each VSI type is represented by a VSI type ID and a VSI version--the network manager can deploy one or more VSI versions at any given time.

- The VM manager configures VSI (which is a virtual station interface for a VM that is represented by a MAC address and VLAN ID pair) . To accomplish this, the VM manager queries available VSI type IDs (VTIDs) and creates a VSI instance consisting of a VSI Instance ID and the chosen VTID. This instance is known as VTDB and contains a VSI manager ID, a VSI type ID, a VSI version, and a VSI instance ID.

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84
- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859

Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch

Virtual machines (VMs) can use a physical switch that is adjacent to the VMs' server to send packets both to other VMs and to the rest of the network when two conditions have been met:

- Virtual Ethernet packet aggregator (VEPA) is configured on the VM server.
- Edge virtual bridging (EVB) is configured on the switch.

This example shows how to configure EVB on the switch so that packets can flow to and from the virtual machines.

Requirements

This example uses the following hardware and software components:

- One EX4500 or EX8200 switch
- Junos OS Release 12.1 or later for EX Series switches

Before you configure EVB on a switch, be sure you have configured the server with virtual machines, the VLANs, and VEPA:

NOTE: **The following are the numbers of components used in this example, but you can use fewer or more to configure the feature.**

- On the server, configure six virtual machines, VM 1 through VM 6 as shown in Figure 55 on page 861. See your server documentation.
- On the server, configure three VLANs named VLAN_Purple, VLAN_Orange, and VLAN_Blue, and add two virtual machines to each VLAN. See your server documentation.
- On the server, install and configure VEPA to aggregate the virtual machine packets.
- On the switch, configure one interface with the same three VLANs as the server (VLAN_Purple, VLAN_Orange, and VLAN_Blue). See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98.

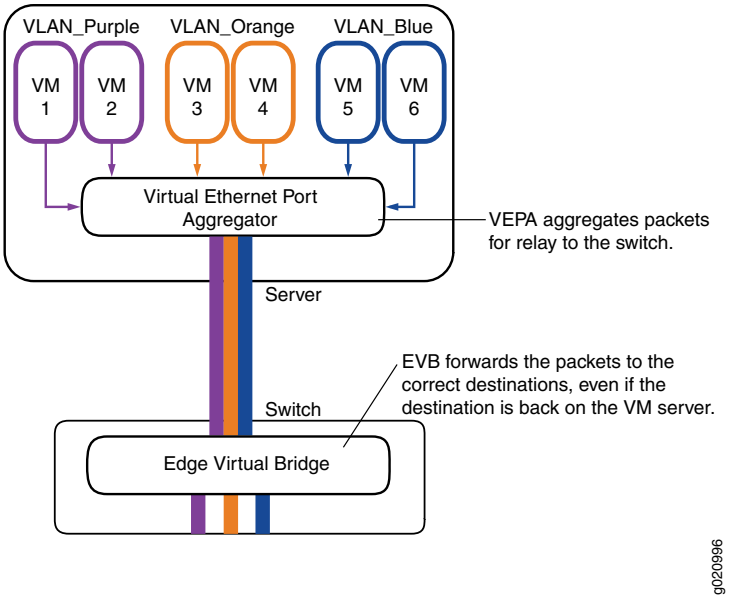
Overview and Topology

EVB is a software capability that provides multiple virtual end stations that communicate with each other and with external switches in the Ethernet network environment.

This example demonstrates the configuration that takes place on a switch when that switch is connected to a server with VEPA configured. In this example, a switch is already connected to a server hosting six virtual machines (VMs) and configured with 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. Because VEPA is configured on the server, no two VMs can communicate directly—all communication between VMs must happen via the adjacent switch. Figure 55 on page 861 shows the topology for this example.

Edge Virtual Bridging Example Topology

Figure 55: Topology



The VEPA component of the server pushes all packets from any VM, regardless of whether the packets are destined to other VMs on the same server or to any external host, to the adjacent switch. The adjacent switch applies policies to incoming packets based on the interface configuration and then forwards the packets to appropriate interfaces based on the MAC learning table. If the switch has not yet learned a destination MAC, it floods the packet to all interfaces, including the source port on which the packet arrived.

Table 118 on page 848 shows the components used in this example.

Table 120: Components of the Topology for Configuring EVB

Component	Description
EX Series switch	For a list of switches that support this feature, see *EX Series Switch Software Features Overview* or *EX Series Virtual Chassis Software Features Overview*.
ge-0/0/20	Switch interface to the server.
Server	Server with virtual machines and VEPA technology.
Virtual machines	Six virtual machines located on the server, named VM 1, VM 2, VM 3, VM 4, VM 5, and VM 6.
VLANs	Three VLANs, named VLAN_Purple, VLAN_Orange, and VLAN_Blue. Each VLAN has two virtual machine members.

Table 120: Components of the Topology for Configuring EVB (continued)

Component	Description
VEPA	A virtual Ethernet port aggregator (VEPA) is a software capability on a server that collaborates with an adjacent, external switch to provide bridging support between multiple virtual machines and with external networks. The VEPA collaborates with the switch by forwarding all VM-originated frames to the adjacent bridge for frame processing and frame relay (including hairpin forwarding) and by steering and replicating frames received from the VEPA uplink to the appropriate destinations.

NOTE: Configuring EVB also enables Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP).

Configuration

CLI Quick Configuration

To quickly configure EVB, copy the following commands and paste them into the switch's CLI at the **[edit]** hierarchy level.

```
set interfaces ge-0/0/20 unit 0 family ethernet-switching port-mode tagged-access
set protocols lldp interface ge-0/0/20.0
set vlans vlan_purple interface ge-0/0/20.0
set vlans vlan_orange interface ge-0/0/20.0
set vlans vlan_blue interface ge-0/0/20.0
set protocols edge-virtual-bridging vsi-discovery interface ge-0/0/20.0
set policy-options vsi-policy P1 from vsi-manager 98 vsi-type 998 vsi-version 4 vsi-instance
09b11c53-8b5c-4eeb-8f00-c84ebb0bb998
set policy-options vsi-policy P1 then filter f2
set policy-options vsi-policy P3 from vsi-manager 97 vsi-type 997 vsi-version 3 vsi-instance
09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
set policy-options vsi-policy P3 then filter f3
set firewall family ethernet-switching filter f2 term t1 then accept
set firewall family ethernet-switching filter f2 term t1 then count f2_accept
set firewall family ethernet-switching filter f3 term t1 then accept
set firewall family ethernet-switching filter f3 term t1 then count f3_accept
set protocols edge-virtual-bridging vsi-discovery vsi-policy P1
set protocols edge-virtual-bridging vsi-discovery vsi-policy P3
```

Step-by-Step Procedure

To configure EVB on the switch:

1. Configure tagged-access mode for the interfaces on which you will enable EVB:

   ```
   [edit interfaces ge-0/0/20]
   user@switch# set unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Enable the Link Layer Discovery Protocol (LLDP) on the ports interfaces on which you will enable EVB:

   ```
   [edit protocols]
   user@switch# set lldp interface ge-0/0/20.0
   ```

3. Configure the interface as a member of all VLANs located on the virtual machines.

   ```
   [edit]
   ```

```
user@switch# set vlans vlan_purple interface ge-0/0/20.0
user@switch# set vlans vlan_orange interface ge-0/0/20.0
user@switch# set vlans vlan_blue interface ge-0/0/20.0
```

4. Enable the VSI Discovery and Control Protocol (VDP) on the interface:

```
[edit protocols]
user@switch# set edge-virtual-bridging vsi-discovery interface ge-0/0/20.0
```

5. Define policies for VSI information. VSI information is based on a VSI manager ID, VSI type, VSI version, and VSI instance ID:

```
[edit policy-options]
user@switch# set vsi-policy P1 from vsi-manager 98 vsi-type 998 vsi-version 4 vsi-instance
09b11c53-8b5c-4eeb-8f00-c84ebb0bb998
user@switch# set vsi-policy P1 then filter f2
user@switch# set vsi-policy P3 from vsi-manager 97 vsi-type 997 vsi-version 3 vsi-instance
09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
user@switch# set vsi-policy P3 then filter f3
```

6. Two VSI policies were defined in the previous step, each of them mapping to different firewall filters. Define the firewall filters:

```
[edit firewall family ethernet-switching]
user@switch# set filter f2 term t1 then accept
user@switch# set filter f2 term t1 then count f2_accept
user@switch# set filter f3 term t1 then accept
user@switch# set filter f3 term t1 then count f3_accept
```

7. Associate VSI policies with VSI-discovery protocol

```
[edit]
user@switch# set protocols edge-virtual-bridging vsi-discovery vsi-policy P1
user@switch# set protocols edge-virtual-bridging vsi-discovery vsi-policy P3
```

Results

```
user@switch# show protocols
edge-virtual-bridging {
    vsi-discovery {
        interface {
            ge-0/0/20.0;
        }
        vsi-policy {
            P1;
            P3;
        }
    }
}
lldp {
    interface ge-0/0/20.0;


user@switch# show policy-options
vsi-policy P1 {
    from {
        vsi-manager 98 vsi-type 998 vsi-version 4 vsi-instance 09b11c53-8b5c-4ee
b-8f00-c84ebb0bb998;
    }
    then {
        filter f2;
    }
}
vsi-policy P3 {
    from {
        vsi-manager 97 vsi-type 997 vsi-version 3 vsi-instance 09b11c53-8b5c-4ee
b-8f00-c84ebb0bb997;
    }
    then {
        filter f3;
    }
}


user@switch# show vlans
vlan_blue {
    interface {
        ge-0/0/20.0;
    }
}
vlan_orange {
    interface {
        ge-0/0/20.0;
    }
}
vlan_purple {
    interface {
        ge-0/0/20.0;
        interface;
    }
}


user@switch# show firewall
family ethernet-switching {
        filter f2 {
        term t1 {
            then {
```

```
                accept;
                count f2_accept;
            }
        }
    }
    filter f3 {
        term t1 {
            then {
                accept;
                count f3_accept;
            }
        }
    }
}
```

Verification

To confirm that EVB is enabled and working correctly, perform these tasks:

- Verifying That EVB is Correctly Configured on page 865
- Verifying That the Virtual Machine Successfully Associated With the Switch on page 865
- Verifying That VSI Profiles Are Being Learned at the Switch on page 866

Verifying That EVB is Correctly Configured

Purpose Verify that EVB is correctly configured

Action

```
user@switch# show edge-virtual-bridging
Interface      Forwarding Mode      RTE   Number of VSIs  Protocols
ge-0/0/20.0    Reflective-relay     25    400             ECP, VDP, RTE
```

Meaning When LLDP is first enabled, an EVB LLDP exchange takes place between switch and server using LLDP. As part of this exchange the following parameters are negotiated: Number of VSIs supported, Forwarding mode, ECP support, VDP support, and Retransmission Timer Exponent (RTE). If the output has values for the negotiated parameters, EVB is correctly configured.

Verifying That the Virtual Machine Successfully Associated With the Switch

Purpose Verify that the virtual machine successfully associated with the switch. After successful association of VSI Profile with the switch interface, verify the learning of the VM's MAC address on MAC-Table or Forwarding database Table. The learn type of the VM's MAC addresses will be VDP, and upon successful shutdown of VM the corresponding MAC-VLAN entry will get flushed out from FDB table otherwise it will never shutdown.

Action

```
user@switch# run show ethernet-switching table
Ethernet-switching table: 10 entries, 4 learned
VLAN MAC address      Type  Age  Interfaces
v3  *        Flood        -      All-members
v3      00:02:a6:11:bb:1a    VDP          -      ge-1/0/10.0
v3      00:02:a6:11:cc:1a    VDP          -      ge-1/0/10.0
v3  00:23:9c:4f:70:01    Static       -      Router
v4      *                    Flood        -      All-members
v4      00:02:a6:11:bb:bb    VDP          -      ge-1/0/10.0
v4      00:23:9c:4f:70:01    Static       -      Router
v5      *                    Flood        -      All-members
v5      00:23:9c:4f:70:01    Static       -      Router
v5      52:54:00:d5:49:11    VDP          -      ge-1/0/20.0
```

Verifying That VSI Profiles Are Being Learned at the Switch

Purpose Verify that VSI profiles are being learned at the switch.

Action

```
user@switch# show edge-virtual-bridging vsi-profiles
Interface: ge-0/0/20.0
Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
      MAC                       VLAN
      00:10:94:00:00:04         3
```

Meaning Whenever VMs configured for VEPA are started at the server, the VMs start sending VDP messages. As part of this protocol VSI profiles are learned at the switch.

If the output has values for Manager, Type, Version, VSI State, and Instance, VSI profiles are being learned at the switch.

Related Documentation

- Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure) on page 867
- Understanding Edge Virtual Bridging for Use with VEPA Technology on EX Series Switches on page 857

Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure)

Configure edge virtual bridging (EVB) when a switch is connected to a virtual machine (VM) server using virtual Ethernet port aggregator (VEPA) technology. EVB does not convert packets; rather, it ensures that packets from one VM destined for another VM on the same VM server is switched. In other words, when the source and destination of a packet are the same port, EVB delivers the packet properly, which otherwise would not happen.

NOTE: Configuring EVB also enables Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP).

Before you begin configuring EVB, ensure that you have:

- Configured packet aggregation on the server connected to the port that you will use on the switch for EVB. See your server documentation.
- Configured the EVB interface for all VLANs located on the virtual machines. See "Configuring VLANs for EX Series Switches (CLI Procedure)" on page 98.

NOTE: The port security features MAC move limiting and MAC limiting are supported on interfaces that are configured for EVB; however, the port security features IP source guard, dynamic ARP inspection (DAI), and DHCP snooping are not supported by EVB. For more information about these features, see *Understanding Port Security Features to Protect the Access Ports on Your Device Against the Loss of Information and Productivity* .

To configure EVB on the switch:

1. Configure tagged-access mode for the interfaces on which you will enable EVB:

   ```
   [edit interfaces interface-name]
   user@switch# set unit 0 family ethernet-switching port-mode tagged-access
   ```

2. Enable the Link Layer Discovery Protocol (LLDP) on the interfaces on which you will enable EVB:.

   ```
   [edit protocols]
   user@switch# set lldp interface interface-name
   ```

3. Configure the interfaces for EVB as members of all VLANs located on the virtual machines.

   ```
   [edit protocols]
   user@switch# set vlans vlan-name vlan-id vlan-number
   ```

4. Enable VDP on the interfaces:

```
[edit protocols]
user@switch# set edge-virtual-bridging vsi-discovery interface interface-name
```

5. Define policies for VSI information, including a VSI manager ID, VSI type, VSI version, and VSI instance ID:

```
[edit policy-options]
user@switch# set vsi-policy policy-name from vsi-manager manager-number vsi-type
type-number vsi-version version-number vsi-instance instance-number
user@switch# set vsi-policy policy-name then filter filter-name
```

6. Define the firewall filters you mapped to in the previous step. When each incoming packet matches the filter, the count is incremented by 1. Other possible actions are accept and drop.

```
[edit firewall family ethernet-switching]
user@switch# set filter filter-name term term-name then action
```

7. Associate VSI policies with VDP:

```
[edit protocols]
user@switch# set edge-virtual-bridging vsi-discovery vsi-policy policy-name
```

8. Verify that the virtual machine successfully associated with the switch. After successful association of the VSI Profile with the switch interface, verify the learning of the VM's MAC address on MAC-Table or Forwarding database Table. The learn type of the VM's MAC addresses will be VDP, and upon successful shutdown of VM the corresponding MAC-VLAN entry will get flushed out from FDB table otherwise it will never shutdown.

```
admin@host# run show ethernet-switching table
```

9. Verify that VSI profiles are being learned at the switch:

```
user@switch# show edge-virtual-bridging vsi-profiles
```

10. Check the statistics of ECP packet exchanges between the switch and server:

```
user@switch# show edge-virtual-bridging ecp statistics
```

Related Documentation

- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859
- Understanding Edge Virtual Bridging for Use with VEPA Technology on EX Series Switches on page 857

CHAPTER 40

Configuring Unknown Unicast Forwarding

Understanding Unknown Unicast Forwarding

Unknown unicast packets have unknown destination MAC addresses. By default, these packets are flooded out all the ports in the VLAN, which can trigger a traffic storm and cause security issues and performance degradation. To prevent storms, you can configure a VLAN to forward unknown unicast traffic to a trunk port, which prevents the packets from being flooded.

Related Documentation

- *Understanding Storm Control*
- *Example: Configuring Storm Control to Prevent Network Outages*
- Configuring Unknown Unicast Forwarding on page 869

Configuring Unknown Unicast Forwarding

Unknown unicast packets have unknown destination MAC addresses. By default, these packets are flooded out all the ports in the VLAN, which can trigger a traffic storm and cause security issues and performance degradation.

To prevent storms, configure a VLAN to forward unknown unicast traffic to a trunk port. The destination MAC address can be learned from the trunk port and added to the Ethernet switching table. You can configure different VLANs to forward unknown unicast traffic to different trunk interfaces or use one trunk interface for multiple VLANs.

To configure unknown unicast forwarding options using the CLI:

NOTE: **Before you can configure unknown unicast forwarding within a VLAN, you must first configure that VLAN.**

1. Configure unknown unicast forwarding for a specific VLAN:

```
[edit ethernet-switching-options]
user@switch# set unknown-unicast-forwarding vlan VLAN-name
```

2. Specify the trunk interface to which unknown unicast traffic will be forwarded:

```
[edit ethernet-switching-options ]
user@switch# set unknown-unicast-forwarding vlan VLAN-name interface (Unknown Unicast Forwarding) interface-name
```

Related Documentation

- Understanding Unknown Unicast Forwarding on page 869
- *Understanding Storm Control*
- *Example: Configuring Storm Control to Prevent Network Outages*
- Configuring VLANs on Switches on page 93

CHAPTER 41

Troubleshooting Ethernet Switching

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 60 to 1,000,000 seconds.)

   ```
   [edit protocols l2-learning]
   user@switch# set global-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 891
- global-mac-table-aging-time on page 974

Troubleshooting Ethernet Switching on EX Series Switches

Troubleshooting issues for Ethernet switching on EX Series switches:

- MAC Address in the Switch's Ethernet Switching Table Is Not Updated After a MAC Address Move on page 872

MAC Address in the Switch's Ethernet Switching Table Is Not Updated After a MAC Address Move

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. However, sometimes silent devices, such as SYSLOG servers or SNMP Trap receivers that receive UDP traffic but do not return acknowledgement (ACK) messages to the traffic source, do not 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. In Junos OS Release 9.4 and later, 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, thereby refreshing the entries in the switch's ARP cache table and Ethernet switching table

Related Documentation

- arp on page 891
- mac-table-aging-time on page 1034

Troubleshooting Private VLANs on QFX Switches

Use the following information to troubleshoot a private VLAN configuration.

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.
- If you want to change a primary VLAN to be a secondary VLAN, you must first change it to a normal VLAN and commit the change. For example, you would follow this procedure:

  1. Change the primary VLAN to be a normal VLAN.
  2. Commit the configuration.
  3. Change the normal VLAN to be a secondary VLAN.
  4. Commit the configuration.

  Follow the same sequence of commits if you want to change a secondary VLAN to be a primary VLAN. That is, make the secondary VLAN a normal VLAN and commit that change and then change the normal VLAN to be a primary VLAN.

Forwarding with Private VLANs

Problem **Description:**

- When isolated VLAN or community VLAN tagged traffic is received on a PVLAN trunk port, MAC addresses are learned from the primary VLAN. This means that output from the show ethernet-switching table command shows that MAC addresses are learned from the primary VLAN and replicated to secondary VLANs. This behavior has no effect on forwarding decisions.
- If a packet with a secondary VLAN tag is received on a promiscuous port, it is accepted and forwarded.
- If a packet is received on a PVLAN trunk port and meets both of the conditions listed below, it is dropped.
  - The packet has a community VLAN tag.
  - The packet is destined to a unicast MAC address or multicast group MAC address that was learned on an isolated VLAN.
- If a packet is received on a PVLAN trunk port and meets both of the conditions listed below, it is dropped.
  - The packet has an isolated VLAN tag.
  - The packet is destined to a unicast MAC address or multicast group MAC address that was learned on a community VLAN.
- If a packet with a primary VLAN tag is received by a secondary (isolated or community) VLAN port, the secondary port forwards the packet.
- If you configure a community VLAN on one device and configure another community VLAN on a second device and both community VLANs use the same VLAN ID, traffic for one of the VLANs can be forwarded to the other VLAN. For example, assume the following configuration:
  - Community VLAN comm1 on switch 1 has VLAN ID 50 and is a member of primary VLAN pvlan100.
  - Community VLAN comm2 on switch 2 also has VLAN ID 50 and is a member of primary VLAN pvlan200.
  - Primary VLAN pvlan100 exists on both switches.

  If traffic for comm1 is sent from switch 1 to switch 2, it will be sent to the ports participating in comm2. (The traffic will also be forwarded to the ports in comm1, as you would expect.)

Solution These are expected behaviors.

Egress Firewall Filters with Private VLANs

Problem **Description:** 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 that carries an isolated VLAN 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 secondary VLAN trunk port that carries a community VLAN 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).

Solution These are expected behaviors. They occur only if you apply a firewall filter to a private VLAN in the output direction and do not occur if you apply a firewall filter to a private VLAN in the input direction.

See Also
- Understanding Private VLANs on page 226

Egress Port Mirroring with Private VLANs

Problem **Description:** If you create a port-mirroring configuration that mirrors private VLAN (PVLAN) traffic on egress, the mirrored traffic (the traffic that is sent to the analyzer system) has the VLAN tag of the ingress VLAN instead of the egress VLAN. For example, assume the following PVLAN configuration:

- Promiscuous trunk port that carries primary VLANs pvlan100 and pvlan400.
- Isolated access port that carries secondary VLAN isolated200. This VLAN is a member of primary VLAN pvlan100.
- Community port that carries secondary VLAN comm300. This VLAN is also a member of primary VLAN pvlan100.
- Output interface (monitor interface) that connects to the analyzer system. This interface forwards the mirrored traffic to the analyzer.

If a packet for pvlan100 enters on the promiscuous trunk port and exits on the isolated access port, the original packet is untagged on egress because it is exiting on an access port. However, the mirror copy retains the tag for pvlan100 when it is sent to the analyzer.

Here is another example: If a packet for comm300 ingresses on the community port and egresses on the promiscuous trunk port, the original packet carries the tag for pvlan100 on egress, as expected. However, the mirrored copy retains the tag for comm300 when it is sent to the analyzer.

Solution This is expected behavior.

Related Documentation

- Understanding Private VLANs on page 226
- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 244
- Creating a Private VLAN on a Single QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch on page 342

PART 2

Configuration Statements and Operational Commands

CHAPTER 42

Configuration Statements

address

List of Syntax

Syntax MX Series and EX Series (dynamic-profiles) on page 885
Syntax QFX Series and QFabric (interfaces) on page 885

Syntax MX Series and EX Series (dynamic-profiles)

```
address (ip-address | ipv6-address);
```

Syntax QFX Series and QFabric (interfaces)

```
address address {
  arp ip-address (mac | multicast-mac) mac-address <publish>;
  broadcast address;
  destination address;
  destination-profile name;
  reui-64;
  master-only;
  multipoint-destination addressdlci dlci-identifier;
  multipoint-destination address {
    epd-threshold cells;
    inverse-arp;
    oam-liveness {
      up-count cells;
      down-count cells;
    }
    oam-period (disable | seconds);
    shaping {
      (cbr rate |rtvbr peak rate sustained rate burst length |vbr peak rate sustained rate burst
        length);
      queue-length number;
    }
    vci vpi-identifier.vci-identifier;
  }
  primary;
  preferred;
  (vrrp-group | vrrp-inet6-group) group-number {
    (accept-data | no-accept-data);
    advertise–interval seconds;
    authentication-type authentication;
    authentication-key key;
    fast-interval milliseconds;
    (preempt | no-preempt) {
      hold-time seconds;
    }
    priority-number number;
    track {
      priority-cost seconds;
      priority-hold-time interface-name {
        interface priority;
        bandwidth-threshold bits-per-second {
          priority;
        }
      }
      route ip-address/mask routing-instance instance-name priority-cost cost;
    }
```

```
    virtual-address [ addresses ];
  }
}
```

MX Series and EX Series (dynamic-profiles)

[edit dynamic-profiles *profile-name* interfaces *interface-name* unit *logical-unit-number* family *family*],
[edit dynamic-profiles *profile-name* interfaces demux0 unit *logical-unit-number* family *family*],
[edit dynamic-profiles *profile-name* interfaces pp0 unit "$junos-interface-unit" family *family*],
[edit interfaces *interface-name* unit *logical-unit-number* family inet],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family *family*]

QFX Series and QFabric (interfaces)

[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*]

Release Information

Statement introduced in Junos OS Release 9.2.
Support at the **[edit dynamic-profiles *profile-name* interfaces pp0 unit "$junos-interface-unit" family *family*]** hierarchy level introduced in Junos OS Release 10.1.
Statement introduced before Junos OS Release 11.1 for QFX Series switches.
Support at the **[edit interfaces *interface-name* unit *logical-unit-number* family *inet*]** hierarchy level introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description

Configure the interface address.

Options

ip-address—IPv4 address of the interface.

ipv6-address—IPv6 address of the interface. When configuring an IPv6 address on a dynamically created interface, use the ***$junos-ipv6–address*** dynamic variable.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring the Protocol Family*
- *Format for Specifying IP Addresses, Network Masks, and Prefixes in Junos OS Configuration Statements*

add-attribute-length-in-pdu

Syntax add-attribute-length-in-pdu;

Hierarchy Level [edit protocols mvrp]

Release Information Statement introduced in Junos OS Release 11.3 for EX Series switches.

Description Add an extra byte in protocol data units (PDUs) sent by the Multiple VLAN Registration Protocol (MVRP) in Junos OS Releases 11.3 and later for EX Series switches that do not support the Enhanced Layer 2 Software (ELS). By default, this MVRP version does not include the extra byte. You can add the extra byte in this MVRP version to address an incompatibility issue with the following MVRP versions:

- MVRP in Junos OS Releases 11.2 and earlier, which includes the extra byte.
- MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for ELS, which includes the extra byte.

If this incompatibility issue arises, the MVRP versions that include the extra byte do not recognize PDUs that do not include the extra byte.

You can recognize an MVRP version problem by looking at a switch running an MVRP version that includes the extra byte. Because a switch running an MVRP version that includes the extra byte cannot interpret an unmodified PDU from an MVRP version that does not include the extra byte, the switch will not add VLANs from the MVRP version that does not include the extra byte. When you execute the command **show mvrp statistics** on the MVRP version that includes the extra byte, the values for *Join Empty received* and *Join In received* will incorrectly display zero, even though the value for *MRPDU received* has been increased. 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 MVRP version that includes the extra byte.

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) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496

aggregated-ether-options

List of Syntax

Syntax (EX, MX Series) on page 888
Syntax (NFX, QFX Series, EX4600, OCX1100, QFabric) on page 888

Syntax (EX, MX Series)

```
aggregated-ether-options {
  ethernet-switch-profile {
    tag-protocol-id;
  }
  (flow-control | no-flow-control);
  lacp {
    (active | passive);
    admin-key key;
    periodic interval;
    system-id mac-address;
  }
  (link-protection | no-link-protection);
  link-speed speed;
  local-bias;
  logical-interface-fpc-redundancy;
  (loopback | no-loopback);
  mc-ae {
    chassis-id chassis-id;
    events {
      iccp-peer-down {
        force-icl-down;
        prefer-status-control-active;
      }
    }
    init-delay-time seconds;
    mc-ae-id mc-ae-id;
    mode (active-active | active-standby);
    redundancy-group group-id;
    revert-time revert-time;
    status-control (active | standby);
    switchover-mode (non-revertive |revertive);
  }
  minimum-links number;
  system-priority
}
```

Syntax (NFX, QFX Series, EX4600, OCX1100, QFabric)

The **fcoe-lag** and **mc-ae** statements are not supported on OCX Series switches.

```
aggregated-ether-options {
  configured-flow-control {
    rx-buffers (on | off);
    tx-buffers (on | off);
  }
  ethernet-switch-profile {
  tag-protocol-id;
  (fcoe-lag | no-fcoe-lag);
  (flow-control | no-flow-control);
  lacp mode {
```

```
    admin-key key;
    periodic interval;
    system-id mac-address;
    force-up;
    }
  }
  (link-protection | no-link-protection);
  link-speed speed;
  local-bias;
  local-minimum-links-threshold threshold-value;
  (loopback | no-loopback);
  mc-ae {
    chassis-id chassis-id;
    mc-ae-id mc-ae-id;
    mode (active-active);
    status-control (active | standby);
  }
  minimum-links number;
  rebalance-periodic;
  resilient-hash;
  source-address-filter filter;
  (source-filtering | no-source-filtering);
}
```

Hierarchy Level (EX Series, QFX Series) [edit interfaces ae*x*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3R2.
Statements **fcoe-lag** and **no-fcoe-lag** introduced in Junos OS Release 13.2X52-D10 for the QFX Series.
Statements **force-up**, **lacp**, and **resilient-hash** introduced in Junos OS Release 14.1X53-D10 for the QFX Series.
Statement **local-minimum-links-threshold** introduced in Junos OS Release 14.1X53-D40 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the aggregated Ethernet properties of a specific aggregated Ethernet interface.

NOTE:

- The **fcoe-lag** and **mc-ae** statements are not supported on OCX Series switches.
- The **force-up** statement is not supported on QFX10002 switches.
- The **resilient-hash** statement is not supported on QFX10002 switches.

The remaining statements are explained separately. See CLI Explorer.

Default Options are not enabled.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Aggregated Ethernet Interfaces and LACP for Switches*
- *Configuring Aggregated Ethernet LACP (CLI Procedure)*
- *Example: Configuring Link Aggregation with LACP Between a QFX Series Product and an Aggregation Switch*
- *Junos OS Network Interfaces Library for Routing Devices*
- *Example: Configuring Aggregated Ethernet High-Speed Uplinks Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*
- *Example: Configuring Aggregated Ethernet High-Speed Uplinks with LACP Between an EX4200 Virtual Chassis Access Switch and an EX4200 Virtual Chassis Distribution Switch*
- *Configuring Aggregated Ethernet Links (CLI Procedure)*
- *Configuring Aggregated Ethernet LACP (CLI Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces for Switches*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- *Junos OS Ethernet Interfaces Configuration Guide*

arp (Interfaces)

Syntax

```
arp ip-address (mac | multicast-mac) mac-address publish;
```

```
arp {
  aging-timer minutes;
  gratuitous-arp-delayseconds;
  gratuitous-arp-on-ifup;
  interfaces {
    interface-name {
      aging-timer minutes;
    }
  }
  passive-learning;
  purging;
}
```

Syntax (EX Series)

```
arp {
  aging-timer minutes;
}
```

Hierarchy Level

[edit system]

[edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*],

[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family inet address *address*]

NOTE: The **edit logical-systems** hierarchy is not available on QFabric systems.

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.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces only, configure Address Resolution Protocol (ARP) table entries, mapping IP addresses to MAC addresses. You can enable backup VRRP routers to learn ARP requests for VRRP-IP to VRRP-MAC address translation. You can also set the time interval between ARP updates.

NOTE: By default, an ARP policer is installed that is shared among all the Ethernet interfaces on which you have configured the **family inet** statement. By including the **arp** statement at the **[edit interfaces *interface-name* unit *logical-unit-number* family inet policer]** hierarchy level, you can apply a specific ARP-packet policer to an interface. This feature is not available on EX Series switches.

When you need to conserve IP addresses, you can configure an Ethernet interface to be unnumbered by including the **unnumbered-address** statement at the **[edit interfaces** *interface-name* **unit** *logical-unit-number* **family inet]** hierarchy level.

NOTE: For EX-Series switches, set only the time interval between ARP updates.

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, **0000.5e00.5355** or **00:00:5e:00:53:55**.

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, **0000.5e00.5355** or **00:00:5e:00:53:55**.

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: For unicast MAC addresses only, if you include the **publish** option, the router or switch replies to proxy ARP requests.

aging-timer—Time interval in minutes between ARP updates. In environments where the number of ARP entries to update is high (for example, on routers only, metro Ethernet environments), increasing the time between updates can improve system performance.

passive-learning (QFX-Series only)—Configure backup VRRP routers or switches to learn the ARP mappings (IP-to-MAC address) for hosts sending the requests. By default, the backup VRRP router drops these requests; therefore, if the master router fails, the backup router must learn all entries present in the ARP cache of the master router. Configuring passive learning reduces transition delay when the backup router is activated.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Configuring Static ARP Table Entries For Mapping IP Addresses to MAC Addresses*
- *Configuring Junos OS ARP Learning and Aging Options for Mapping IPv4 Network Addresses to MAC Addresses*
- *Junos OS Network Interfaces Library for Routing Devices*
- *Junos OS System Basics Configuration Guide* .

autostate-exclude

Syntax autostate-exclude;

Hierarchy Level [edit interface *interface-name* ether-options]

Release Information Statement introduced in Junos OS Release 14.1x53-D40 and Junos OS Release 17.3R1 on QFX5100 switches.

Description Specify not to include an IRB interface in the state calculation for VLAN members. The default behavior is not to exclude an IRB interface in the state calculation unless all the ports on the interface go down. Because an IRB interface often has multiple ports in a single VLAN, the state calculation for a VLAN member might include a port that is down, possibly resulting in traffic loss. This feature enables you to exclude a trunk or access interface from the state calculation, which results in the IRB interface being marked as down as soon as the port specifically assigned to a VLAN goes down.

IRB interfaces are used to bind specific VLANs to Layer 3 interfaces, enabling a switch to forward packets between those VLANs— without having to configure another device, such as a router, to connect VLANs. In a typical scenario, a port on the interface is assigned to a specific VLAN, while a different port on that interface is assigned to an 802.1Q trunk interface to carry traffic between multiple VLANs, and a third port on that interface is assigned to an access interface used to connect the VLAN to network devices.

To ensure that an interface is marked as down and thereby excluded from the state calculation for VLAN members when the port assigned to the VLAN goes down, configure this statement on the trunk or access interface. The trunk or port interface is automatically excluded from the state calculation of the IRB interface. In this way, when a port assigned to a specified VLAN goes down, the IRB interface assigned to that VLAN is also marked as down.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration

Related Documentation

- Excluding an IRB Interface from State Calculations on a QFX Series Switch on page 468
- port-mode on page 1072
- show ethernet-switching interface on page 1212

bpdu-destination-mac-address

Syntax bpdu-destination-mac-address provider-bridge-group;

MX Series and EX Series
[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit protocols mvrp],
[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type)

SRX Series
[edit protocols mvrp],
[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type)

Release Information
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description For Multiple VLAN Registration Protocol (MVRP) configurations, specifies the multicast address for MVRP. If configured, the provider MVRP multicast MAC address is used; otherwise, the Junos OS uses the customer MVRP multicast MAC address.

Default By default, the provider MVRP multicast MAC address is used (if configured). Otherwise, the customer MVRP MAC address is used.

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 MX Series Routers*
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on Security Devices on page 513
- Verifying That MVRP Is Working Correctly on page 551
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501

bridge-priority

Syntax bridge-priority *priority*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols (mstp | rstp)],
[edit logical-systems *logical-system-name* protocols mstp msti *msti-id*],
[edit logical-systems *logical-system-name* protocols vstp vlan *vlan-id*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mstp msti *msti-id*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*],

[edit routing-instances *routing-instance-name* protocols (mstp | rstp)],
[edit routing-instances *routing-instance-name* protocols mstp msti *msti-id*],
[edit routing-instances *routing-instance-name* protocols vstp vlan *vlan-id*]

[edit protocols mstp],
[edit protocols mstp msti *msti-id*],
[edit protocols rstp],
[edit protocols stp],
[edit protocols vstp vlan *vlan-id*]

Release Information
Statement introduced in Junos OS Release 8.4.
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.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description Configures the bridge priority, which 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.

Default 32,768

Options ***priority***—The bridge priority 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

- *Understanding MSTP*
- *Understanding VSTP*
- *Understanding Bridge Priority for Election of Root Bridge and Designated Bridge*
- *Example: Configuring Network Regions for VLANs with MSTP on Switches*

- *Example: Configuring Network Regions for VLANs with MSTP on non-ELS EX Series Switches*
- *show spanning-tree bridge*
- *show spanning-tree interface*

code-points (CoS)

Syntax code-points ([*aliases*] | [*bit-patterns*]);

Hierarchy Level [edit class-of-service classifiers *type classifier-name* forwarding-class *class-name* loss-priority *level*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.2 for SRX Series devices.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.1X44 for the SRX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Statement introduced in Junos OS Release 14.2 for PTX Series Packet Transport Routers.

Description Specify one or more DSCP code-point aliases or bit sets to apply to a forwarding class..

NOTE: OCX Series switches do not support MPLS, and therefore, do not support EXP code points or code point aliases.

Options ***aliases***—Name of the DSCP alias.

bit-patterns—Value of the code-point bits, in six-bit binary form.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Interfaces*
- *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*
- *Example: Configuring Behavior Aggregate Classifiers*
- Example: Configuring BA Classifiers on Transparent Mode Security Devices on page 665

community-vlan

Syntax community-vlan vlan *community-vlan-name*;

Hierarchy Level [edit vlans *primary-vlan-name* vlan-id *primary-vlan-vlan-id*]

Release Information Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.

Description Configure the specified community VLAN to be a secondary VLAN of the specified primary VLAN. A *community* VLAN is 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.

NOTE: Before you specify this configuration statement, you must have already configured the specified community VLAN and assigned a VLAN ID to it. See private-vlan.

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 with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277

control-channel

Syntax

```
control-channel channel-name {
  vlan vlan-id;
  interface name interface-name
}
```

Hierarchy Level [edit protocols protection-group ethernet-ring *name* (east-interface | west-interface)]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description Configure the Ethernet RPS control channel logical interface to carry the RAPS PDU. The related physical interface is the physical ring port.

Options

vlan ***vlan-id***—If the control channel logical interface is a trunk port, then a dedicated **vlan** ***vlan-id*** defines the dedicated VLAN channel to carry the RAPS traffic. Only configure the ***vlan-id*** when the control channel logical interface is the trunk port.

interface name ***interface-name***—Interface name of the control channel.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

control-vlan

Syntax control-vlan *(vlan-id | vlan-name)*

Hierarchy Level [edit protocols protection-group ethernet-ring]

[edit protocols protection-group ethernet-ring name (east-interface |west-interface)]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description Specify the VLAN that carries the protocol data units (PDUs) between the nodes in the protected Ethernet ring. This is a control VLAN, meaning that it carries data for one instance of an Ethernet ring protection switching (ERPS) in the control channel. Use a control VLAN on trunk port interfaces. One control channel can contain multiple control VLANs.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

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 on page 914
- ether-type on page 932
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554

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 Forwarding Mode on Switches on page 30

data-channel

Syntax

```
data-channel {
  vlan number;
}
```

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information

Statement introduced in Junos OS Release 10.2.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

For Ethernet ring protection, configure a data channel to define a set of VLAN IDs that belong to a ring instance.

VLANs specified in the data channel use the same topology used by the ERPS PDU in the control channel. Therefore, if a ring interface is blocked in the control channel, all traffic in the data channel is also blocked on that interface.

Options

vlan ***number***—Specify (by VLAN ID) one or more VLANs that belong to a ring instance.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- *Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers*
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

description (Interfaces)

Syntax description *text*;

Hierarchy Level [edit interfaces *interface-name*],
[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 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Provide a textual description of the interface or the logical unit. Any descriptive text you include is displayed in the output of the **show interfaces** commands, and is also exposed in the **ifAlias** Management Information Base (MIB) object. It has no effect on the operation of the interface on the router or switch.

The textual description can also be included in the extended DHCP relay option 82 Agent Circuit ID suboption.

Options ***text***—Text to describe the interface. If the text includes spaces, enclose the entire text in quotation marks.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Interface Description*
- *Adding a Logical Unit Description to the Configuration*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit and 10-Gigabit Ethernet Interfaces for OCX Series Switches*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure) for EX Series Switches with ELS support*
- *Configuring Gigabit and 10-Gigabit Ethernet Interfaces for EX4600 and QFX Series Switches*
- *Using DHCP Relay Agent Option 82 Information*
- *Junos OS Network Interfaces Library for Routing Devices*
- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support on page 170

description (VLAN)

Syntax description *text-description*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Option **text-description** enhanced from supporting up to 128 characters to supporting up to 256 characters in Junos OS Release 10.2 for EX Series switches.

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.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84
- Understanding Bridging and VLANs on Switches on page 84
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch with ELS Support on page 131
- **show vlans on page 1510**

destination-address (Security Policies)

Syntax

```
destination-address {
    [address];
    any;
    any-ipv4;
    any-ipv6;
}
```

Hierarchy Level

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* match]

[edit security policies global policy *policy-name* match]

Release Information

Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 10.2. Support for IPv6 addresses in active/active chassis cluster configurations (in addition to the existing support of active/passive chassis cluster configurations) added in Junos OS Release 10.4. Support for wildcard addresses added in Junos OS Release 11.1.

Description

Define the matching criteria. You can specify one or more IP addresses, address sets, or wildcard addresses. You can specify wildcards **any**, **any-ipv4**, or **any-ipv6**.

Options

address—IP address (**any**, **any-ipv4**, **any-ipv6**), IP address set, or address book entry, or wildcard address (represented as A.B.C.D/wildcard-mask). You can configure multiple addresses or address prefixes separated by spaces and enclosed in square brackets.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Security Policies Overview*

dhcp-relay

Syntax

```
dhcp-relay {
  active-server-group server-group-name;
  authentication {
    password password-string;
    username-include {
      circuit-type;
      delimiter delimiter-character;
      domain-name domain-name-string;
      interface-name;
      logical-system-name;
      mac-address;
      option-60;
      option-82 <circuit-id> <remote-id>;
      routing-instance-name;
      user-prefix user-prefix-string;
    }
  }
  dhcpv6 {
    active-server-group server-group-name;
    authentication {
    password password-string;
      username-include {
        circuit-type;
        client-id;
        delimiter delimiter-character;
        domain-name domain-name-string;
        interface-name;
        logical-system-name;
        relay-agent-interface-id;
        relay-agent-remote-id;
        relay-agent-subscriber-id;
        routing-instance-name;
        user-prefix user-prefix-string;
      }
    dynamic-profile profile-name {
      aggregate-clients (merge | replace);
      use-primary primary-profile-name;
    }
    }
  }
    group group-name {
      active-server-group server-group-name;
      authentication {
        ...
      }
      dynamic-profile profile-name {
        ...
      }
      interface interface-name {
        exclude;
        liveness-detection {
          failure-action (clear-binding | clear-binding-if-interface-up | log-only);
```

```
        method {
          bfd {
            version (0 | 1 | automatic);
            minimum-interval milliseconds;
            minimum-receive-interval milliseconds;
            multiplier number;
            no-adaptation;
            transmit-interval {
              minimum-interval milliseconds;
              threshold milliseconds;
            }
            detection-time {
              threshold milliseconds;
            }
            session-mode(automatic | multihop | singlehop);
            holddown-interval milliseconds;
          }
        }
      }
      overrides {
        ...
      }
      service-profile dynamic-profile-name;
      trace;
      upto upto-interface-name;
    }
    service-profile dynamic-profile-name;
    }
    overrides {
      ...
    }
    relay-agent-interface-id {
      ...
    }
  service-profile dynamic-profile-name;
  liveness-detection {
    failure-action (clear-binding | clear-binding-if-interface-up | log-only);
    method {
      bfd {
        version (0 | 1 | automatic);
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        multiplier number;
        no-adaptation;
        transmit-interval {
          minimum-interval milliseconds;
          threshold milliseconds;
        }
        detection-time {
          threshold milliseconds;
        }
        session-mode(automatic | multihop | singlehop);
        holddown-interval milliseconds;
      }
    }
  }
```

```
    overrides {
      allow-snooped-clients;
      interface-client-limit number;
      no-allow-snooped-clients;
      no-bind-on-request;
      send-release-on-delete;
    }
    relay-agent-interface-id {
      prefix prefix;
      use-interface-description (logical | device):
    }
    server-group {
      server-group-name {
        server-ip-address;
      }
    }
  dynamic-profile profile-name {
    aggregate-clients (merge | replace);
    use-primary primary-profile-name;
  }
  forward-snooped-clients (all-interfaces | configured-interfaces |
    non-configured-interfaces);
  group group-name {
    active-server-group server-group-name;
    authentication {
      ...
    }
    dynamic-profile profile-name {
      ...
    }
    interface interface-name {
      exclude;
      liveness-detection {
        failure-action (clear-binding | clear-binding-if-interface-up | log-only);
        method {
          bfd {
            version (0 | 1 | automatic);
            minimum-interval milliseconds;
            minimum-receive-interval milliseconds;
            multiplier number;
            no-adaptation;
            transmit-interval {
              minimum-interval milliseconds;
              threshold milliseconds;
            }
            detection-time {
              threshold milliseconds;
            }
            session-mode(automatic | multihop | singlehop);
            holddown-interval milliseconds;
          }
        }
      }
      overrides {
        ...
      }
```

```
      service-profile dynamic-profile-name;
      trace;
      upto upto-interface-name;
    }
    overrides {
      ...
    }
    relay-option-82 {
      ...
    }
    service-profile dynamic-profile-name;
  }
  liveness-detection {
    failure-action (clear-binding | clear-binding-if-interface-up | log-only);
    method {
      bfd {
        version (0 | 1 | automatic);
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        multiplier number;
        no-adaptation;
        transmit-interval {
          minimum-interval milliseconds;
          threshold milliseconds;
        }
        detection-time {
          threshold milliseconds;
        }
        session-mode(automatic | multihop | singlehop);
        holddown-interval milliseconds;
      }
    }
  }
  overrides {
    allow-snooped-clients;
    always-write-giaddr;
    always-write-option-82;
    client-discover-match <option60-and-option82>;
    disable-relay;
    interface-client-limit number;
    layer2-unicast-replies;
    no-allow-snooped-clients;
    no-bind-on-request;
    proxy-mode;
    replace-ip-source-with;
    send-release-on-delete;
    trust-option-82;
  }
  relay-option-82 {
    circuit-id {
      prefix prefix;
      use-interface-description (logical | device);
    }
  }
  server-group {
    server-group-name {
```

```
        server-ip-address;
      }
    }
    service-profile dynamic-profile-name;
}
```

Hierarchy Level [edit forwarding-options],
[edit vlans forwarding-options]

Release Information Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure extended Dynamic Host Configuration Protocol (DHCP) relay and DHCPv6 relay options on the switch and enable the switch to function as a DHCP relay agent. A DHCP relay agent forwards DHCP request and reply packets between a DHCP client and a DHCP server.

DHCP relay supports the attachment of dynamic profiles and also interacts with the local AAA Service Framework to use back-end authentication servers, such as RADIUS, to provide subscriber authentication. You can attach dynamic profiles and configure authentication support on a global basis or for a specific group of interfaces.

The extended DHCP and DHCPv6 relay agent options configured with the **dhcp-relay** and **dhcpv6** statements are incompatible with the DHCP/BOOTP relay agent options configured with the **bootp** statement. As a result, the extended DHCP or DHCPv6 relay agent and the DHCP/BOOTP relay agent cannot both be enabled on the router at the same time.

The remaining 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

- *Configuring DHCP and BOOTP*

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) on Switches on page 504

domain-type (Bridge Domains)

Syntax domain-type bridge;

ACX Series and MX Series
[edit bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]

SRX Series [edit bridge-domains *bridge-domain-name*]

Release Information
Statement introduced in Junos OS Release 8.4.
Statement modified in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.

Description Define the domain type **bridge** for a Layer 2 bridge domain.

NOTE: There is only one domain type **bridge**, that can be configured on SRX Series devices. Domain type **bridge** is not enabled by default. An SRX Series device operates in the Layer 2 transparent mode when all physical bridge domains on the device are partitioned into logical bridge domains.

NOTE: Starting with Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the CLI **domain-type** is not available.

NOTE: Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, the hierarchy **[edit bridge-domains bridge-domain-name]** is renamed to **[edit vlans vlan-name]**. For detailed information about the modified hierarchies, see "Enhanced Layer 2 CLI Configuration Statement and Command Changes for Security Devices" on page 385.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Layer 2 Transparent Mode Overview on page 377
- *Configuring a Bridge Domain*
- *Configuring a Layer 2 Virtual Switch*

dot1q-tunneling

List of Syntax

Syntax (Ethernet Switching) on page 914
Syntax (VLANs) on page 914

Syntax (Ethernet Switching)

```
dot1q-tunneling {
  ether-type (0x8100 | 0x88a8 | 0x9100);
}
```

Syntax (VLANs)

```
dot1q-tunneling {
  customer-vlans (id | native | range);
  layer2-protocol-tunneling all | protocol-name {
    drop-threshold number;
    shutdown-threshold number;
  }
}
```

Ethernet Switching

[edit ethernet-switching-options]

VLANs

[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

For Ethernet switching, sets a global value for the EtherType for Q-in-Q tunneling.

For VLANs, enables 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. See CLI Explorer.

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 on page 601
- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582

- Configuring Q-in-Q Tunneling on QFX Series Switches on page 581
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598

dot1x

Syntax

```
dot1x {
  authenticator {
    authentication-profile-name access-profile-name;
    interface (all | <interface-names>) {
      disable;
      guest-vlan (vlan-tag | vlan-name);
      lldp-med-bypass;
      mac-radius <restrict | flap-on-disconnect>;
      maximum-requests number;
      no-reauthentication;
      quiet-period seconds;
      reauthentication interval;
      retries retries-number;
      server-fail (deny | permit | use-cache | vlan-name vlan-name );
      server-reject-vlan vlan-name;
      server-timeout seconds;
      supplicant (single | single-secure | multiple);
      supplicant-timeout seconds;
      transmit-period seconds
    }
    no-mac-table-binding;
    use-vlan-name <use-vlan-id | use-vlan-name>;
    static  mac-address ;
  traceoptions {
      file filename  <files number> <no-world-readable | world-readable> <size size>;
      flag;
  }
}
```

Hierarchy Level [edit protocols]

Release Information Command introduced in Junos OS Release 15.1X49-D80.

Description Configure 802.1X authentication for port-based network access control (PNAC). 802.1X authentication is supported on interfaces that are members of private VLANs (PVLANs).

Define tracing operations for the 802.1X protocol authentication.

Default 802.1X authentication is disabled.

Options **authentication-profile-name** *access-profile-name*—Name of the access profile to use for authentication.

all—Configure all interfaces for 802.1X authentication.

interface-names—List of names of interfaces to configure for 802.1X authentication.

guest-vlan *vlan-tag* —VLAN tag identifier of the guest VLAN.

guest-vlan *vlan-name*—Name of the guest VLAN.

mac-radius flap-on-disconnect—(Optional) Reset an interface on receiving a disconnect request.

mac-radius restrict—(Optional) Bypass dot1x authentication. This restricts authentication to MAC RADIUS.

maximum-requests *number*—Maximum number of times an EAPOL request packet is retransmitted to the supplicant before the authentication session times out.

quiet-period *seconds*—Number of seconds the interface remains in the wait state.

reauthentication *interval*—Sets the periodic reauthentication time interval in seconds.

retries *number*—Number of retries after which port is placed into wait state.

deny—Force fail the supplicant authentication. No traffic flows through the interface.

permit—Force succeed the supplicant authentication. Traffic flows through the interface as if it were successfully authenticated by the RADIUS server.

use-cache—Force succeed the supplicant authentication only if it was previously authenticated successfully. This action ensures that already authenticated supplicants are not affected.

server-fail *vlan-tag* —Move supplicant on the interface to the VLAN specified by this numeric identifier.

server-fail *vlan-name*—Move supplicant on the interface to the VLAN specified by this name.

server-fail *seconds*—The time interval, in seconds, during which the device does not attempt to contact the authentication server to reauthenticate a client that has already been authenticated using server fail fallback.

server-timeout *seconds*—Amount of time a port waits for a reply when relaying a response from the supplicant to the authentication server before timing out and invoking the server-fail action.

single—Authenticates only the first client that connects to an authenticator port.

single-secure—Authenticates only one client to connect to an authenticator port.

multiple—Authenticates multiple clients individually on one authenticator port.

supplicant-timeout *seconds*—Number of seconds the port waits for a response when relaying a request from the authentication server to the supplicant before resending the request.

transmit-period *seconds*—Number of seconds the port waits before retransmitting the initial EAPOL PDUs to the supplicant.

mac-address—The MAC address of the device for which 802.1X authentication must be bypassed and the device permitted access to the port.

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.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—All tracing operations.
- **config-internal**—Trace internal configuration operations.
- **eapol**—Trace EAPOL packets transmitted and received.
- **general**—Trace general operations.
- **normal**—Trace normal operations.
- **parse**—Trace reading of the configuration.
- **state**—Trace protocol state changes.
- **task**—Trace protocol task operations.
- **timer**—Trace protocol timer operations.
- **vlan**—Trace VLAN transactions.

no-world-readable—(Optional) Restrict file access to the user who created the file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB).

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

- Understanding 802.1X Port-Based Network Authentication on page 789
- clear dot1x on page 1166

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 on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395
- shutdown-threshold on page 1104

east-interface

Syntax

```
east-interface {
  node-id mac-address;
  control-channel channel-name {
    vlan number;
    interface name interface-name
  }
  interface-none
  ring-protection-link-end;
}
```

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description

Define one of the two interface ports for Ethernet ring protection, the other being defined by the **west-interface** statement at the same hierarchy level. The interface must use the control channel's logical interface name. The control channel is a dedicated VLAN channel for the ring port.

EX Series switches do not use the node-id statement--the node ID is automatically configured on the switches using the MAC address.

NOTE: Always configure this port first, before configuring the **west-interface** statement.

NOTE: The Node ID is not configurable on EX Series switches. The node ID is automatically configured using the MAC address.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- west-interface on page 1160
- ethernet-ring on page 934

- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

edge-virtual-bridging

Syntax

```
edge-virtual-bridging {
  traceoptions {
    file filename <files number> <no-stamp> <replace> <size size> <world-readable |
      no-world-readable>;
    flag flag ;
  }
  vsi-discovery {
    interface interface-name
    vsi-policy vsi-policy-name
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description Configure edge virtual bridging (EVB). EVB enables a virtualized station (a physical end station, a server, connected to virtual machines (VMs)) to network with an adjacent switch so that applications residing on the virtual machines can interact with each other and external networks through a technology called virtual Ethernet packet aggregator (VEPA).

The remaining statements are explained separately. See CLI Explorer.

Default EVB is disabled by default.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859
- Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure) on page 867

enable-all-ifl

Syntax enable-all-ifl;

Hierarchy Level [edit protocols layer2-control mac-rewrite interface *interface-name*]

Release Information Statement introduced in Junos OS Release 13.3.

Description Enable tunneling for STP, VTP, CDP, and other supported protocols on all logical interfaces (VLANs) configured on the interface.

NOTE: Tunneling on all logical interfaces is enabled automatically for PVST/PVST+.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Layer 2 Protocol Tunneling Through a Network*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389
- protocol on page 1084

encapsulation

List of Syntax

Syntax for Physical Interfaces: M Series, MX Series, QFX Series, T Series, PTX Series on page 924
Syntax for Logical Interfaces: SRX Series on page 924

Syntax for Physical Interfaces: M Series, MX Series, QFX Series, T Series, PTX Series

```
encapsulation (atm-ccc-cell-relay | atm-pvc | cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc |
  ethernet-bridge | ethernet-ccc | ethernet-over-atm | ethernet-tcc | ethernet-vpls |
  ethernet-vpls-fr | ether-vpls-over-atm-llc | ethernet-vpls-ppp | extended-frame-relay-ccc
  | extended-frame-relay-ether-type-tcc | extended-frame-relay-tcc | extended-vlan-bridge
  | extended-vlan-ccc | extended-vlan-tcc | extended-vlan-vpls | flexible-ethernet-services
  | flexible-frame-relay | frame-relay | frame-relay-ccc | frame-relay-ether-type |
  frame-relay-ether-type-tcc | frame-relay-port-ccc | frame-relay-tcc | generic-services |
  multilink-frame-relay-uni-nni | ppp | ppp-ccc | ppp-tcc | vlan-ccc | vlan-vci-ccc | vlan-vpls);
```

Syntax for Logical Interfaces: SRX Series

```
encapsulation (ether-vpls-ppp | ethernet-bridge | ethernet-ccc | ethernet-tcc | ethernet-vpls
  | extended-frame-relay-ccc | extended-frame-relay-tcc | extended-vlan-bridge |
  extended-vlan-ccc | extended-vlan-tcc | extended-vlan-vpls | frame-relay-port-ccc |
  vlan-ccc | vlan-vpls);
```

Physical Interfaces: M Series, MX Series, QFX Series, T Series, PTX Series

[edit interfaces *interface-name*],
[edit interfaces rlsq *number:number*]

Logical Interfaces: SRX Series

[edit interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 11.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1X48 for PTX Series Packet Transport Routers (**flexible-ethernet-services**, **ethernet-ccc**, and **ethernet-tcc** options only).

Description

For M Series, MX Series, QFX Series, T Series, PTX Series, specify the physical link-layer encapsulation type.

For SRX Series, specify logical link layer encapsulation.

NOTE: Not all encapsulation types are supported on the switches. See the switch CLI.

Default

ppp—Use serial PPP encapsulation.

Physical Interface Options and Logical Interface Options

[Warning: element unresolved in stylesheets: <title> (in <config-options>). This is probably a new element that is not yet supported in the stylesheets.]
Physical Interface Options and Logical Interface Options

For physical interfaces:

NOTE: **Frame Relay, ATM, PPP, SONET, and SATSOP options are not supported on EX Series switches.**

- **atm-ccc-cell-relay**—Use ATM cell-relay encapsulation.
- **atm-pvc**—Defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you configure physical ATM interfaces with ATM PVC encapsulation, an RFC 2684-compliant ATM Adaptation Layer 5 (AAL5) tunnel is set up to route the ATM cells over a Multiprotocol Label Switching (MPLS) path that is typically established between two MPLS-capable routers using the Label Distribution Protocol (LDP).
- **cisco-hdlc**—Use Cisco-compatible High-Level Data Link Control (HDLC) framing. E1, E3, SONET/SDH, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:
  - CCC version (**cisco-hdlc-ccc**)—The logical interface does not require an encapsulation statement. When you use this encapsulation type, you can configure the **ccc** family only.
  - TCC version (**cisco-hdlc-tcc**)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.
- **cisco-hdlc-ccc**—Use Cisco-compatible HDLC framing on CCC circuits.
- **cisco-hdlc-tcc**—Use Cisco-compatible HDLC framing on TCC circuits for connecting different media.
- **ethernet-bridge**—Use Ethernet bridge encapsulation on Ethernet interfaces that have bridging enabled and that must accept all packets.
- **ethernet-over-atm**—For interfaces that carry IPv4 traffic, use Ethernet over ATM encapsulation. When you use this encapsulation type, you cannot configure multipoint interfaces. As defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, this encapsulation type allows ATM interfaces to connect to devices that support only bridge protocol data units (BPDUs). Junos OS does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and MAC header, and the packet is forwarded to the ATM interface.
- **ethernet-tcc**—For interfaces that carry IPv4 traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. For 8-port, 12-port, and 48-port Fast Ethernet PICs, TCC is not supported.

- **ethernet-vpls**—Use Ethernet VPLS encapsulation on Ethernet interfaces that have VPLS enabled and that must accept packets carrying standard TPID values. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.
- **ethernet-vpls-fr**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 Frame Relay connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use the MAC address to forward the packet into a given VPLS instance.
- **ethernet-vpls-ppp**—Use in a VPLS setup when a CE device is connected to a PE device over a time division multiplexing (TDM) link. This encapsulation type enables the PE device to terminate the outer Layer 2 PPP connection, use the 802.1p bits inside the inner Ethernet header to classify the packets, look at the MAC address from the Ethernet header, and use it to forward the packet into a given VPLS instance.
- **ether-vpls-over-atm-llc**—For ATM intelligent queuing (IQ) interfaces only, use the Ethernet virtual private LAN service (VPLS) over ATM LLC encapsulation to bridge Ethernet interfaces and ATM interfaces over a VPLS routing instance (as described in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*). Packets from the ATM interfaces are converted to standard ENET2/802.3 encapsulated Ethernet frames with the frame check sequence (FCS) field removed.
- **extended-frame-relay-ccc**—Use Frame Relay encapsulation on CCC circuits. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to CCC. When you use this encapsulation type, you can configure the **ccc** family only.
- **extended-frame-relay-ether-type-tcc**—Use extended Frame Relay ether type TCC for Cisco-compatible Frame Relay for DLCIs 1 through 1022. This encapsulation type is used for circuits with different media on either side of the connection.
- **extended-frame-relay-tcc**—Use Frame Relay encapsulation on TCC circuits to connect different media. This encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC.
- **extended-vlan-bridge**—Use extended VLAN bridge encapsulation on Ethernet interfaces that have IEEE 802.1Q VLAN tagging and bridging enabled and that must accept packets carrying TPID 0x8100 or a user-defined TPID.
- **extended-vlan-ccc**—Use extended VLAN encapsulation on CCC circuits with Gigabit Ethernet and 4-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values. Extended VLAN CCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. When you use this encapsulation type, you can configure the **ccc** family only. For 8-port, 12-port, and 48-port Fast Ethernet PICs, extended VLAN CCC is not supported. For 4-port Gigabit Ethernet PICs, extended VLAN CCC is not supported.
- **extended-vlan-tcc**—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. For 4-port Gigabit Ethernet PICs, extended VLAN TCC is not supported.

- **extended-vlan-vpls**—Use extended VLAN VPLS encapsulation on Ethernet interfaces that have VLAN 802.1Q tagging and VPLS enabled and that must accept packets carrying TPIDs 0x8100, 0x9100, and 0x9901. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

NOTE: The built-in Gigabit Ethernet PIC on an M7i router does not support extended VLAN VPLS encapsulation.

- **flexible-ethernet-services**—For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router), and for Gigabit Ethernet interfaces, use flexible Ethernet services encapsulation when you want to configure multiple per-unit Ethernet encapsulations. Aggregated Ethernet bundles can use this encapsulation type. This encapsulation type allows you to configure any combination of route, TCC, CCC, Layer 2 virtual private networks (VPNs), and VPLS encapsulations on a single physical port. If you configure flexible Ethernet services encapsulation on the physical interface, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.
- **flexible-frame-relay**—For IQ interfaces only, use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.
- **frame-relay**—Use Frame Relay encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, link services, SONET/SDH, T1, T3, and voice services interfaces can use Frame Relay encapsulation.
- **frame-relay-ccc**—Use Frame Relay encapsulation on CCC circuits. This encapsulation is same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC. The logical interface must also have **frame-relay-ccc** encapsulation. When you use this encapsulation type, you can configure the **ccc** family only.
- **frame-relay-ether-type**—Use Frame Relay ether type encapsulation for compatibility with the Cisco Frame Relay. IETF frame relay encapsulation identifies the payload format using NLPID and SNAP formats. Cisco-compatible Frame Relay encapsulation uses the Ethernet type to identify the type of payload.

NOTE: When the encapsulation type is set to Cisco-compatible Frame Relay encapsulation, ensure that the LMI type is set to ANSI or Q933-A.

- **frame-relay-ether-type-tcc**—Use Frame Relay ether type TCC for Cisco-compatible Frame Relay on TCC circuits to connect different media. This encapsulation is Cisco-compatible Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to TCC.

- **frame-relay-port-ccc**—Use Frame Relay port CCC encapsulation to transparently carry all the DLCIs between two customer edge (CE) routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. The connection between the two CE routers can be either user-to-network interface (UNI) or network-to-network interface (NNI); this is completely transparent to the PE routers. When you use this encapsulation type, you can configure the **ccc** family only.
- **frame-relay-tcc**—This encapsulation is similar to Frame Relay CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.
- **generic-services**—Use generic services encapsulation for services with a hierarchical scheduler.
- **multilink-frame-relay-uni-nni**—Use MLFR UNI NNI encapsulation. This encapsulation is used on link services, voice services interfaces functioning as FRF.16 bundles, and their constituent T1 or E1 interfaces, and is supported on LSQ and redundant LSQ interfaces.
-
- **ppp**—Use serial PPP encapsulation. This encapsulation is defined in RFC 1661, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET/SDH, T1, and T3 interfaces can use PPP encapsulation.
- **ppp-ccc**—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only.
- **ppp-tcc**—Use serial PPP encapsulation on TCC circuits for connecting different media. When you use this encapsulation type, you can configure the **tcc** family only.
- **vlan-ccc**—Use Ethernet VLAN encapsulation on CCC circuits. VLAN CCC encapsulation supports TPID 0x8100 only. When you use this encapsulation type, you can configure the **ccc** family only.

- **vlan-vci-ccc**—Use ATM-to-Ethernet interworking encapsulation on CCC circuits. When you use this encapsulation type, you can configure the **ccc** family only. All logical interfaces configured on the Ethernet interface must also have the encapsulation type set to **vlan-vci-ccc**.
- **vlan-vpls**—Use VLAN VPLS encapsulation on Ethernet interfaces with VLAN tagging and VPLS enabled. Interfaces with VLAN VPLS encapsulation accept packets carrying standard TPID values only. On M Series routers, except the M320 router, the 4-port Fast Ethernet TX PIC and the 1-port, 2-port, and 4-port, 4-slot Gigabit Ethernet PICs can use the Ethernet VPLS encapsulation type.

NOTE:

- Label-switched interfaces (LSIs) do not support VLAN VPLS encapsulation. Therefore, you can only use VLAN VPLS encapsulation on a PE-router-to-CE-router interface and not a core-facing interface.
- Starting with Junos OS release 13.3, a commit error occurs when you configure **vlan-vpls** encapsulation on a physical interface and configure **family inet** on one of the logical units. Previously, it was possible to commit this invalid configuration.

For logical interfaces:

- **frame-relay**—Configure a Frame Relay encapsulation when the physical interface has multiple logical units, and the units are either point to point or multipoint.
- **multilink-frame-relay-uni-nni**—Link services interfaces functioning as FRF.16 bundles can use Multilink Frame Relay UNI NNI encapsulation.
- **ppp**—For normal mode (when the device is using only one ISDN B-channel per call). Point-to-Point Protocol is for communication between two computers using a serial interface.
- **ppp-over-ether**—This encapsulation is used for underlying interfaces of pp0 interfaces.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Physical Encapsulation on an Interface*
- *Configuring Interface Encapsulation on Physical Interfaces*
- *Configuring CCC Encapsulation for Layer 2 VPNs*
- *Configuring Layer 2 Switching Cross-Connects Using CCC*
- *Configuring TCC Encapsulation for Layer 2 VPNs and Layer 2 Circuits*
- *Configuring ATM Interface Encapsulation*
- *Configuring ATM-to-Ethernet Interworking*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *Configuring VLAN and Extended VLAN Encapsulation*
- *Configuring Encapsulation for Layer 2 Wholesale VLAN Interfaces*
- *Configuring Interfaces for Layer 2 Circuits*
- *Configuring Interface Encapsulation on PTX Series Packet Transport Routers*
- *Configuring MPLS LSP Tunnel Cross-Connects Using CCC*
- *Configuring TCC*
- *Configuring VPLS Interface Encapsulation*
- *Configuring Interfaces for VPLS Routing*
- *Defining the Encapsulation for Switching Cross-Connects*
- *Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)*

ether-options

Syntax

```
ether-options {
  802.3ad {
    aex;
    (backup | primary);
    lacp {
      force-up;
      port-priority
    }
  }
  (auto-negotiation | no-auto-negotiation);
  ethernet-switch-profile {
    tag-protocol-id;
  }
  (flow-control | no-flow-control);
  ieee-802-3az-eee;
  link-mode mode;
  (loopback | no-loopback);
  speed (speed | auto-negotiation);
}
```

Hierarchy Level

[edit interfaces *interface-name*],
[edit interfaces interface-range *range*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.3R2.

Description

Configure Ethernet properties for a Gigabit Ethernet interface or a 10-Gigabit Ethernet interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure) for EX Series Switches with ELS support*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- *Configuring LACP Link Protection of Aggregated Ethernet Interfaces for Switches*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- *Junos OS Ethernet Interfaces Configuration Guide*

ether-type

Syntax ether-type (0x8100 | 0x88a8 | 0x9100)

Hierarchy Level [edit ethernet-switching-options dot1q-tunneling]

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 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. The Ethertype value appears in the Ethernet type field of the packet. It specifies the protocol being transported in the Ethernet frame.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- dot1q-tunneling on page 914
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582
- Configuring Q-in-Q Tunneling on QFX Series Switches on page 581
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598

ethernet (Chassis Cluster)

Syntax

```
ethernet {
  device-count number;
  lacp {
    link-protection {
      non-revertive;
    }
    system-priority number;
  }
}
```

Hierarchy Level [edit chassis aggregated-devices]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure properties for aggregated Ethernet devices.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

ethernet-ring

Syntax

```
ethernet-ring ring-name {
  control-vlan (vlan-id | vlan-name);
  data-channel {
    vlan number
  }
  east-interface {
    control-channel channel-name {
      vlan number;
      interface name interface-name
    }
  }
  guard-interval number;
  node-id mac-address;
  restore-interval number;
  ring-protection-link-owner;
  west-interface {
    control-channel channel-name {
      vlan number;
    }
  }
}
```

Hierarchy Level [edit protocols protection-group]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description For Ethernet PICs on MX Series routers or for EX Series switches, , specify the Ethernet ring in an Ethernet ring protection switching configuration.

Options *ring-name*—Name of the Ethernet protection ring.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

ethernet-switch-profile

Syntax

```
ethernet-switch-profile {
  ethernet-policer-profile {
    input-priority-map {
      ieee802.1p premium [values];
    }
    output-priority-map {
      classifier {
        premium {
          forwarding-class class-name {
            loss-priority (high | low);
          }
        }
      }
    }
    policer cos-policer-name {
      aggregate {
        bandwidth-limit bps;
        burst-size-limit bytes;
      }
      premium {
        bandwidth-limit bps;
        burst-size-limit bytes;
      }
    }
    storm-control storm-control-profile;
    tag-protocol-id tpid;
  }
  mac-learn-enable;
}
```

Hierarchy Level

[edit interfaces *interface-name* gigether-options],
[edit interfaces *interface-name* aggregated-ether-options],
[edit interfaces *interface-name* aggregated-ether-options],
[edit interfaces *interface-name* ether-options]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 13.2X50-D15 for the EX Series switches.

Description

NOTE: On QFX Series standalone switches, the ethernet-policer-profile CLI hierarchy and the mac-learn-enable statement are supported only on the Enhanced Layer 2 Switching CLI.

For Gigabit Ethernet IQ, 10-Gigabit Ethernet IQ2 and IQ2-E, and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC, aggregated Ethernet with Gigabit Ethernet IQ interfaces, the built-in Gigabit Ethernet port on the M7i router); 100-Gigabit

Ethernet Type 5 PIC with CFP; and Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces on EX Series switches, configure VLAN tag and MAC address accounting and filtering properties.

The remaining statements are explained separately. See CLI Explorer.

NOTE: When you gather interfaces into a bridge domain, the **no-mac-learn-enable** statement at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile]** hierarchy level is not supported. You must use the **no-mac-learning** statement at the **[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*]** hierarchy level to disable MAC learning on an interface in a bridge domain. For information on disabling MAC learning for a bridge domain, see the *MX Series Layer 2 Configuration Guide*.

Default If the **ethernet-switch-profile** statement is not configured, Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i router) behave like Gigabit Ethernet interfaces.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Gigabit Ethernet Policers*
- *Configuring MAC Address Filtering*
- *Stacking and Rewriting Gigabit Ethernet VLAN Tags Overview*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

ethernet-switching

List of Syntax

Syntax (EX Series and QFX Series) on page 937
Syntax (SRX Series) on page 937

Syntax (EX Series and QFX Series)

```
ethernet-switching {
  filter input filter-name;
  filter output filter-name;
  native-vlan-id vlan-id;
  port-mode mode;
  reflective-relay;
  vlan {
    members [ (all | names | vlan-ids) ];
  }
}
```

Syntax (SRX Series)

```
ethernet-switching {
  block-non-ip-all;
  bpdu-vlan-flooding;
  bypass-non-ip-unicast;
  no-packet-flooding {
    no-trace-route;
  }
}
```

Hierarchy Level

For EX Series and QFX Series switches:

[edit interfaces ge-*chassis/slot/port* unit *logical-unit-number*] family

For SRX Series devices:

[edit security flow]

Release Information

Statement introduced in Junos OS Release 9.5 for SRX Series.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure Ethernet switching protocol family information for the logical interface. Changes default Layer 2 forwarding behavior.

The remaining statements are explained separately. See CLI Explorer.

Default

You must configure a logical interface to be able to use the physical device.

Options

- **block-non-ip-all**—Block all Layer 2 non-IP and non-ARP traffic, including multicast and broadcast traffic.
- **bypass-non-ip-unicast**—Allow all Layer 2 non-IP traffic to pass through the device.
- **no-packet-flooding**—Stop IP flooding and send ARP or ICMP requests to discover the destination MAC address for a unicast packet.

NOTE: On all SRX Series devices in transparent mode, packet flooding is enabled by default. If you have manually disabled packet flooding with the set security flow ethernet-switching no-packet-flooding command, then multicast packets such as OSPFv3 hello packets are dropped.

- **no-trace-route**—Do not send ICMP requests to discover the destination MAC address for a unicast packet. Only ARP requests are sent. This option only allows the device to discover the destination MAC address for a unicast packet if the destination IP address is in the same subnetwork as the ingress IP address.

NOTE: The **block-non-ip-all** and **bypass-non-ip-unicast** options cannot be configured at the same time.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.
security—To view this in the configuration.
security-control—To add this to the configuration.

Related Documentation

- *Configuring Gigabit and 10-Gigabit Ethernet Interfaces for EX4600 and QFX Series Switches*
- *Understanding Traffic Processing on Security Devices*
- *JUNOS Software Network Interfaces Configuration Guide*

ethernet-switching-options

List of Syntax

EX Series on page 939
QFX Series, QFabric, EX4600 on page 942

EX Series

```
ethernet-switching-options {
  analyzer (Port Mirroring) {
    name {
      loss-priority priority;
      ratio number;
      input {
        ingress {
          interface (all | interface-name);
          vlan (vlan-id | vlan-name);
        }
        egress {
          interface (all | interface-name);
        }
      }
      output {
        interface interface-name;
        vlan (vlan-id | vlan-name) {
        no-tag;
        }
      }
    }
  }
  bpdu-block {
    disable-timeout timeout;
    interface (all | [interface-name]) {
      (disable | drop | shutdown);
      }
  }
  dot1q-tunneling {
    ether-type (0x8100 | 0x88a8 | 0x9100);
  }
  interfaces interface-name {
    no-mac-learning;
  }
  mac-lookup-length number-of-entries;
}
  mac-notification {
    notification-interval seconds;
  }
  mac-table-aging-time seconds;
  nonstop-bridging;
  port-error-disable {
    disable-timeout timeout;
  }
  redundant-trunk-group {
    group name {
      interface interface-name <primary>;
      interface interface-name;
    }
```

```
}
secure-access-port {
  dhcp-snooping-file {
    location local_pathname | remote_URL;
    timeout seconds;
    write-interval seconds;
  }
  dhcpv6-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 (drop | log | none | shutdown);
    no-allowed-mac-log;
    persistent-learning;
    static-ip ip-address {
      vlan  vlan-name;
      mac mac-address;
    }
    static-ipv6 ip-address {
      vlan  vlan-name;
      mac mac-address;
    }
  }
  vlan (all | vlan-name) {
    (arp-inspection | no-arp-inspection) [
      forwarding-class class-name;
    }
    dhcp-option82 {
      circuit-id {
        prefix hostname;
        use-interface-description;
        use-vlan-id;
      }
      remote-id {
        prefix hostname | mac | none;
        use-interface-description;
        use-string string;
      }
      vendor-id [string];
    }
    (examine-dhcp | no-examine-dhcp) {
      forwarding-class class-name;
    }
    (examine-dhcpv6 | no-examine-dhcpv6) {
      forwarding-class class-name;
    }
    examine-fip {
      fc-map fc-map-value;
    }
```

```
    (ip-source-guard | no-ip-source-guard);
    (ipv6-source-guard | no-ipv6-source-guard);
    mac-move-limit limit action (drop | log | none | shutdown);
    }
    (neighbor-discovery-inspection | no-neighbor-discovery-inspection);
  no-option-37;
  static {
    vlan name {
      mac mac-address {
        next-hop interface-name;
      }
    }
  }
  storm-control {
    action-shutdown;
    interface (all | interface-name) {
      bandwidth bandwidth;
      level level;
      multicast;
      no-broadcast;
      no-multicast;
      no-registered-multicast;
      no-unknown-unicast;
      no-unregistered-multicast;
    }
  }
  traceoptions {
    file filename <files number> <no-stamp> <replace> <size size> <world-readable |
      no-world-readable>;
    flag flag <disable>;
  }
  unknown-unicast-forwarding {
    vlan (all | vlan-name) {
      interface interface-name;
    }
  }
  voip {
    interface (all | [interface-name | access-ports]) {
      vlan vlan-name;
      forwarding-class (assured-forwarding | best-effort | expedited-forwarding |
        network-control);
    }
  }
}
```

QFX Series, QFabric, EX4600

```
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;
        }
        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 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure Ethernet switching options.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Understanding Port Mirroring*
- *Overview of Access Port Protection*
- *Understanding Port Security Features to Protect the Access Ports on Your Device Against the Loss of Information and Productivity*
- *Understanding BPDU Protection for STP, RSTP, and MSTP*
- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610
- *Understanding Storm Control*
- *Understanding Storm Control on EX Series Switches*
- *Understanding 802.1X and VoIP on EX Series Switches*
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- *Understanding Unknown Unicast Forwarding*
- Understanding MAC Notification on EX Series Switches on page 73
- *Understanding FIP Snooping*
- *Understanding Nonstop Bridging on EX Series Switches*

exclusive-mac

Syntax exclusive-mac *virtual-mac-mac-address/mask*;

Hierarchy Level [edit protocols l2-learning global-mac-move]

Release Information Statement introduced in Junos OS Release 14.1X53-D45.

Description Exclude MAC addresses from the MAC move limit algorithm.

The global MAC move feature is used to track MAC addresses when they appear on a different physical interface or within a different unit of the same physical interface. When you configure the **exclusive-mac *virtual-mac-mac-address/mask*** parameter at the [**edit protocols l2-learning global-mac-move**] hierarchy level, specified MAC addresses are excluded and will not be tracked.

This feature can be useful in OVSDB-managed topologies with VRRP servers deployed in a redundancy configuration (master/slave), and when MAC move limit is configured. Both servers could negotiate mastership, and the same MAC address could be learned under the global MAC move feature while negotiation is occurring. In such cases, excluding the MAC address of the VRRP servers by using the **exclusive-mac** statement prevents this “false” move from being tracked.

The following example excludes VRRP V2 virtual router MAC addresses, as defined in RFC 3768:

```
[edit]
set protocols l2-learning global-mac-move exclusive-mac 00:00:5e:00:01:00/40
```

The following example excludes VRRP V3 virtual router MAC addresses, as defined in RFC 5798:

```
[edit]
set protocols l2-learning global-mac-move exclusive-mac 00:00:5e:00:02:00/40
```

Options ***virtual-mac-mac-address/mask***— Specify a MAC address and a mask. If the mask is 48, only the exact MAC address is excluded. If the mask is 40, all the MAC addresses that have the same first 5 bytes are excluded.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Configuring MAC Move Parameters on page 54

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 244
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch on page 342

fabric-control

Syntax

```
fabric-control {
  graceful-restart {
    restart-timeseconds;
    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. See CLI Explorer.

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*

family

Syntax

```
family family {
  accounting {
    destination-class-usage;
    source-class-usage {
      (input | output | input output);
    }
  }
  access-concentrator name;
  address address {
    ... the address subhierarchy appears after the main [edit interfaces interface-name unit
      logical-unit-number family family-name] hierarchy ...
  }
  bundle interface-name;
  core-facing;
  demux-destination {
    destination-prefix;
  }
  demux-source {
    source-prefix;
  }
  direct-connect;
  duplicate-protection;
  dynamic-profile profile-name;
  filter {
    group filter-group-number;
    input filter-name;
    input-list [  filter-names ];
    output filter-name;
    output-list [  filter-names ];
  }
  interface-mode (access | trunk);
  ipsec-sa sa-name;
  keep-address-and-control;
  mac-validate (loose | strict);
  max-sessions number;
  max-sessions-vsa-ignore;
  mtu bytes;
  multicast-only;
  negotiate-address;
  no-redirects;
  policer {
    arp policer-template-name;
    input policer-template-name;
    output policer-template-name;
  }
  primary;
  protocols [inet iso mpls];
  proxy inet-address address;
  receive-options-packets;
  receive-ttl-exceeded;
  remote (inet-address address | mac-address address);
  rpf-check {
```

```
    fail-filter filter-name
    mode loose;
}
sampling {
    input;
    output;
}
service {
    input {
        post-service-filter filter-name;
        service-set service-set-name <service-filter filter-name>;
    }
    output {
        service-set service-set-name <service-filter filter-name>;
    }
}
service-name-table table-name;
short-cycle-protection < lockout-time-min minimum-seconds lockout-time-max
    maximum-seconds> <filter [aci]>;
(translate-discard-eligible | no-translate-discard-eligible);
(translate-fecn-and-becn | no-translate-fecn-and-becn);
translate-plp-control-word-de;
unnumbered-address interface-name destination address destination-profile profile-name;
vlan-id number;
vlan-id-list [number number-number];
address address {
    arp ip-address (mac | multicast-mac) mac-address <publish>;
    broadcast address;
    destination address;
    destination-profile name;
    eui-64;
    master-only;
    multipoint-destination address dlci dlci-identifier;
    multipoint-destination address {
        epd-threshold cells;
        inverse-arp;
        oam-liveness {
            up-count cells;
            down-count cells;
        }
        oam-period (disable | seconds);
        shaping {
            (cbr rate | rtvbr burst length peak rate sustained rate | vbr burst length peak rate
                sustained rate);
            queue-length number;
        }
        vci vpi-identifier.vci-identifier;
    }
    preferred;
    primary;
    vrrp-groupgroup-id {
        (accept-data | no-accept-data);
        advertise-interval seconds;
        authentication-key key;
        authentication-type authentication;
        fast-interval milliseconds;
```

```
        (preempt | no-preempt) {
          hold-time seconds;
        }
        priority number;
        track {
          interface interface-name {
            bandwidth-threshold bits-per-second priority-cost priority;
            priority-cost priority;
          }
          priority-hold-time seconds;
          route prefix routing-instance instance-name priority-cost priority;
          }
        }
        virtual-address [ addresses ];
        }
      virtual-link-local-address ipv6-address;
    }
}
```

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.
Option **max-sessions-vsa-ignore** introduced in Junos OS Release 11.4.

Description Configure protocol family information for the logical interface.

NOTE: Not all subordinate statements are available to every protocol family.

Options *family*— Protocol family:

- **any**—Protocol-independent family used for Layer 2 packet filtering

NOTE: This option is not supported on T4000 Type 5 FPCs.

- **bridge**—(M Series and T Series routers only) Configure only when the physical interface is configured with **ethernet-bridge** type encapsulation or when the logical interface is configured with **vlan-bridge** type encapsulation. You can optionally configure this protocol family for the logical interface on which you configure VPLS.
- **ethernet-switching**—(M Series and T Series routers only) Configure only when the physical interface is configured with **ethernet-bridge** type encapsulation or when the logical interface is configured with **vlan-bridge** type encapsulation
- **ccc**—Circuit cross-connect protocol suite. You can configure this protocol family for the logical interface of CCC physical interfaces. When you use this encapsulation type, you can configure the **ccc** family only.
- **inet**—Internet Protocol version 4 suite. You must configure this protocol family for the logical interface to support IP protocol traffic, including Open Shortest Path First (OSPF), Border Gateway Protocol (BGP), Internet Control Message Protocol (ICMP), and Internet Protocol Control Protocol (IPCP).
- **inet6**—Internet Protocol version 6 suite. You must configure this protocol family for the logical interface to support IPv6 protocol traffic, including Routing Information Protocol for IPv6 (RIPng), Intermediate System-to-Intermediate System (IS-IS), BGP, and Virtual Router Redundancy Protocol for IPv6 (VRRP).
- **iso**—International Organization for Standardization Open Systems Interconnection (ISO OSI) protocol suite. You must configure this protocol family for the logical interface to support IS-IS traffic.
- **mlfr-end-to-end**—Multilink Frame Relay FRF.15. You must configure this protocol or multilink Point-to-Point Protocol (MLPPP) for the logical interface to support multilink bundling.
- **mlfr-uni-nni**—Multilink Frame Relay FRF.16. You must configure this protocol or **mlfr-end-to-end** for the logical interface to support link services and voice services bundling.
- **multilink-ppp**—Multilink Point-to-Point Protocol. You must configure this protocol (or **mlfr-end-to-end**) for the logical interface to support multilink bundling.
- **mpls**—Multiprotocol Label Switching (MPLS). You must configure this protocol family for the logical interface to participate in an MPLS path.
- **pppoe**—Point-to-Point Protocol over Ethernet
- **tcc**—Translational cross-connect protocol suite. You can configure this protocol family for the logical interface of TCC physical interfaces.

- **tnp**—Trivial Network Protocol. This protocol is used to communicate between the Routing Engine and the router's packet forwarding components. The Junos OS automatically configures this protocol family on the router's internal interfaces only, as discussed in *Understanding Internal Ethernet Interfaces*.
- **vpls**—(M Series and T Series routers only) Virtual private LAN service. You can optionally configure this protocol family for the logical interface on which you configure VPLS. VPLS provides an Ethernet-based point-to-multipoint Layer 2 VPN to connect customer edge (CE) routers across an MPLS backbone. When you configure a VPLS encapsulation type, the **family vpls** statement is assumed by default.

  MX Series routers support dynamic profiles for VPLS pseudowires, VLAN identifier translation, and automatic bridge domain configuration.

  For more information about VPLS, see the *Junos OS VPNs Library for Routing Devices* .

The remaining statements are explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

interface— To view this statement in the configuration.
interface-control— To add this statement to the configuration.

Related Documentation

- *Configuring the Protocol Family*

family inet (Interfaces)

Syntax

```
inet {
  accounting {
    destination-class-usage;
    source-class-usage {
      input;
      output;
    }
  }
  address (source–address/prefix) {
    arp destination-address {
      (mac mac-address | multicast-mac multicast-mac-address);
      publish publish-address;
    }
    broadcast address;
    preferred;
    primary;
    vrrp-group group-id {
      (accept-data | no-accept-data);
      advertise-interval seconds;
      advertisements-threshold number;
      authentication-key key-value;
      authentication-type (md5 | simple);
      fast-interval milliseconds;
      inet6-advertise-interval milliseconds
      (preempt <hold-time seconds> | no-preempt );
      priority value;
      track {
        interface interface-name {
          bandwidth-threshold bandwidth;
          priority-cost value;
        }
        priority-hold-time seconds;
        route route-address{
          routing-instance routing-instance;
          priority-cost value;
        }
      }
      virtual-address [address];
      virtual-link-local-address address;
      vrrp-inherit-from {
        active-group value;
        active-interface interface-name;
      }
    }
    web-authentication {
      http;
      https;
      redirect-to-https;
    }
  }
  dhcp {
    client-identifier {
```

```
      (ascii string | hexadecimal string);
    }
    lease-time (length | infinite);
    retransmission-attempt value;
    retransmission-interval seconds;
    server-address server-address;
    update-server;
    vendor-id vendor-id ;
  }
  dhcp-client {
    client-identifier {
      prefix {
        host-name;
        logical-system-name;
        routing-instance-name;
      }
      use-interface-description (device | logical);
      user-id (ascii string| hexadecimal string);
    }
    lease-time (length | infinite);
    retransmission-attempt value;
    retransmission-interval seconds;
    server-address server-address;
    update-server;
    vendor-id vendor-id ;
  }
  filter {
    group number;
    input filter-name;
    input-list [filter-name];
    output filter-name;
    output-list [filter-name];
  }
  mtu value;
  no-neighbor-learn;
  no-redirects;
  policer {
    arp arp-name;
    input input-name;
    output output-name;
  }
  primary;
  rpf-check {
    fail-filter filter-name;
    mode {
      loose;
    }
  }
  sampling {
    input;
    output;
    simple-filter;
  }
  targeted-broadcast {
    (forward-and-send-to-re |forward-only);
  }
```

```
    unnumbered-address {
      interface-name;
      preferred-source-address preferred-source-address;
    }
}
```

Hierarchy Level [edit interfaces *interface* unit *unit*]

Release Information Statement supported in Junos 10.2 for SRX Series devices.

Description Assign an IP address to a logical interface.

Options *ipaddress*—Specify the IP address for the interface. The remaining statements are explained separately.

NOTE: You use family inet to assign an IPv4 address. You use family inet6 to assign an IPv6 address. An interface can be configured with both an IPv4 and IPv6 address.

Required Privilege Level **interface**—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Interfaces*

family inet6

Syntax

```
inet6 {
  accounting {
    destination-class-usage;
    source-class-usage {
      input;
      ouput;
    }
  }
  address source–address/prefix {
    eui-64;
    ndp address {
      (mac mac-address | multicast-mac multicast-mac-address);
      publish;
    }
    preferred;
    primary;
    vrrp-inet6-group group_id {
      (accept-data | no-accept-data);
      advertisements-threshold number;
      authentication-key value;
      authentication-type (md5 | simple);
      fast-interval milliseconds;
      inet6-advertise-interval milliseconds;
      (preempt <hold-time seconds>| no-preempt );
      priority value;
      track {
        interface interface-name {
          bandwidth-threshold value;
          priority-cost value;
        }
        priority-hold-time seconds;
        route route-address{
          routing-instance routing-instance;
        }
      }
      virtual-inet6-address [address];
      virtual-link-local-address address;
      vrrp-inherit-from {
        active-group value;
        active-interface interface-name;
      }
    }
    web-authentication {
      http;
      https;
      redirect-to-https;
    }
  }
  (dad-disable | no-dad-disable);
  dhcpv6-client {
    client-ia-type (ia-na | ia-pd);
    client-identifier duid-type (duid-ll | duid-llt | vendor);
```

```
    client-type (autoconfig | stateful);
    rapid-commit;
    req-option (dns-server | domain | fqdn | nis-domain | nis-server | ntp-server | sip-domain
      | sip-server |time-zone | vendor-spec);
    retransmission-attempt number;
    update-router-advertisement {
      interface interface-name;
    }
    update-server;
  }
  filter {
    group number;
    input filter-name;
    input-list [filter-name];
    output filter-name;
    output-list [filter-name];
  }
  mtu value;
  nd6-stale-time seconds;
  no-neighbor-learn;
  policer {
    input input-name;
    output output-name;
  }
  rpf-check {
    fail-filter filter-name;
    mode {
      loose;
    }
  }
  sampling {
    input;
    output;
  }
  unnumbered-address {
    interface-name;
    preferred-source-address preferred-source-address;
  }
  ndp-proxy | dad-proxy {
    interface-restricted
    }
}
```

Hierarchy Level [edit interfaces *interface* unit *unit*]

Release Information Statement supported in Junos 10.2 for SRX Series devices.

Description Assign an IPV6 address to a logical interface.

Options *ipaddress*—Specify the IP address for the interface. The remaining statements are explained separately.

NOTE: You use family **inet6** to assign an IPv6 address. You use family **inet** to assign an IPv4 address. An interface can be configured with both an IPv4 and IPv6 address.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *Understanding Interfaces*

fast-aps-switch

Syntax fast-aps-switch;

Hierarchy Level [edit interfaces *interface-name* sonet-options aps]

Release Information Statement introduced in Junos OS Release 12.1.

Description (M320 routers with Channelized OC3/STM1 Circuit Emulation PIC with SFP only, EX Series switches, and MX series routers with Channelized OC3/STM1 Circuit Emulation PIC with SFP only using container interfaces) Reduce the Automatic Protection Switching (APS) switchover time in Layer 2 circuits.

NOTE:

- The fast APS switching feature is supported only within a single chassis on a MX series router using a container interface.
- Configuring this statement reduces the APS switchover time only when the Layer 2 circuit encapsulation type for the interface receiving traffic from a Layer 2 circuit neighbor is SAToP.
- When the **fast-aps-switch** statement is configured in revertive APS mode, you must configure an appropriate value for revert time to achieve reduction in APS switchover time.
- To prevent the logical interfaces in the data path from being shut down, configure appropriate hold-time values on all the interfaces in the data path that support TDM.
- The **fast-aps-switch** statement cannot be configured when the APS **annex-b** option is configured.
- The interfaces that have the **fast-aps-switch** statement configured cannot be used in virtual private LAN service (VPLS) environments.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Reducing APS Switchover Time in Layer 2 Circuits*

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 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description Apply a firewall filter to traffic entering or exiting a VLAN.

Default All incoming traffic is accepted unmodified to the VLAN, and all outgoing traffic is sent unmodified from the VLAN.

Options ***filter-name*** —Name of a firewall filter defined in a **filter** statement.

- **input**—Apply a firewall filter to VLAN ingress traffic.
- **output**—Apply a firewall filter to VLAN egress traffic.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Example: Configuring Firewall Filters for Port, VLAN, and Router Traffic on EX Series Switches*
- *Configuring Firewall Filters*
- *Configuring Firewall Filters (CLI Procedure)*
- *Overview of Firewall Filters*
- *Firewall Filters for EX Series Switches Overview*
- Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure) on page 102

flexible-vlan-tagging

Syntax flexible-vlan-tagging;

Hierarchy Level [edit interfaces ae*x*],
[edit interfaces ge-*fpc/pic/port*],
[edit interfaces et-*fpc/pic/port*],
[edit interfaces ps0],
[edit interfaces xe-*fpc/pic/port*]

Release Information Statement introduced in Junos OS Release 8.1.
Support for aggregated Ethernet added in Junos OS Release 9.0.
Statement introduced in Junos OS Release 12.1x48 for PTX Series Packet Transport Routers.
Statement introduced in Junos OS Release 13.2X50-D15 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description Support simultaneous transmission of 802.1Q VLAN single-tag and dual-tag frames on logical interfaces on the same Ethernet port, and on pseudowire logical interfaces.

This statement is supported on M Series and T Series routers, for Fast Ethernet and Gigabit Ethernet interfaces only on Gigabit Ethernet IQ2 and IQ2-E, IQ, and IQE PICs, and for aggregated Ethernet interfaces with member links in IQ2, IQ2-E, and IQ PICs or in MX Series DPCs, or on Ethernet interfaces for PTX Series Packet Transport Routers or 100-Gigabit Ethernet Type 5 PIC with CFP.

This statement is supported on Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces on EX Series and QFX Series switches.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Enabling VLAN Tagging*
- *Configuring Flexible VLAN Tagging on PTX Series Packet Transport Routers*
- Configuring Double-Tagged VLANs on Layer 3 Logical Interfaces on page 46

flow (Security Flow)

Syntax

```
flow {
  aging {
    early-ageout seconds;
    high-watermark percent;
    low-watermark percent;
  }
  allow-dns-reply;
  ethernet-switching {
    block-non-ip-all;
    bpdu-vlan-flooding;
    bypass-non-ip-unicast;
    no-packet-flooding {
      no-trace-route;
    }
  }
  force-ip-reassembly;
  ipsec-performance-acceleration;
  load distribution {
    session-affinity ipsec;
  }
  packet-log {
    enable;
    throttle-interval;
    packet-filter <filter-name>;
  }
  pending-sess-queue-length (high | moderate | normal);
  route-change-timeout seconds;
  syn-flood-protection-mode (syn-cookie | syn-proxy);
  tcp-mss {
    all-tcp mss value;
    gre-in {
      mss value;
    }
    gre-out {
      mss value;
    }
    ipsec-vpn {
      mss value;
    }
  }
  tcp-session {
    fin-invalidate-session;
    no-sequence-check;
    no-syn-check;
    no-syn-check-in-tunnel;
    rst-invalidate-session;
    rst-sequence-check;
    strict-syn-check;
    tcp-initial-timeout seconds;
    time-wait-state {
      (session-ageout | session-timeout seconds);
    }
```

```
    }
    traceoptions {
      file {
        filename;
        files number;
        match regular-expression;
        size maximum-file-size;
        (world-readable | no-world-readable);
      }
      flag flag;
      no-remote-trace;
      packet-filter filter-name {
        destination-port port-identifier;
        destination-prefix address;
        interface interface-name;
        protocol protocol-identifier;
        source-port port-identifier;
        source-prefix address;
      }
      rate-limit messages-per-second;
    }
}
```

Hierarchy Level [edit security]

Release Information Statement modified in Junos OS Release 9.5.

Description Determine how the device manages packet flow. The device can regulate packet flow in the following ways:

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Understanding Traffic Processing on Security Devices*
- *Understanding Session Characteristics for SRX Series Services Gateways*
- *Understanding Packet Flow in Logical Systems for SRX Series Devices*

forwarding-classes (CoS)

Syntax

```
forwarding-classes {
  class class-name {
    priority (high | low);
    queue-num number;
    spu-priority (high | low | medium-high | medium-low);
  }
  queue queue-number {
    class-name {
      priority (high | low);
    }
  }
}
```

Hierarchy Level [edit class-of-service]

Release Information Statement introduced in Junos OS Release 8.5. Statement updated in Junos OS Release 11.4. The **spu-priority** option introduced in Junos OS Release 11.4R2.

Description Configure forwarding classes and assign queue numbers.

Options

- **class class-name**—Display the forwarding class name assigned to the internal queue number.

  NOTE: This option is supported only on SRX1500, SRX5400, SRX5600, and SRX5800.

  NOTE: AppQoS forwarding classes must be different from those defined for interface-based rewriters.

- **priority**—Fabric priority value:
  - **high**—Forwarding class' fabric queuing has high priority.
  - **low**—Forwarding class' fabric queuing has low priority.

  The default **priority** is **low**.
- **queue queue-number**—Specify the internal queue number to which a forwarding class is assigned.
- **spu-priority**—Services Processing Unit (SPU) priority queue, **high**, **medium-high**, **medium-low**, or **low**. The default **spu-priority** is **low**.

NOTE: The **spu-priority** option is only supported on SRX1500 devices and SRX5000 line devices.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Example: Configuring AppQoS*

forwarding-options

Syntax

```
forwarding-options {
  dhcp-security {
    arp-inspection;
    group group-name {
      interface interface-name {
        static-ip ip-address {
          mac mac-address;
        }
      }
      overrides {
        no-option82;
        (trusted | untrusted);
      }
    }
    ip-source-guard;
    no-dhcp-snooping;
    option-82 {
      circuit-id {
        prefix {
          host-name;
          logical-system-name;
          routing-instance-name;
        }
        use-interface-description (device | logical);
        use-vlan-id;
      }
      remote-id {
        host-name hostname;
        use-interface-description (device | logical);
        use-string string;
      }
      vendor-id {
        use-string string;
      }
    }
  }
  filter (VLANs) {
    input filter-name;
    output filter-name;
  }
  flood {
    input filter-name;
  }
```

Chassis: EX4600 and QFX Series

```
forwarding options profile-name {
  num-65-127-prefix number;
}
```

Chassis: EX4600 and QFX Series

```
forwarding-options lpm-profile {
  prefix-65-127-disable;
  unicast-in-lpm;
```

```
}
```

Chassis: EX4600 and QFX Series

```
forwarding-options custom-profile {
  l2-entries | l3-entries | lpm-entries {
    num-banks number;
  }
}
```

Hierarchy Level

[edit],
[edit bridge-domains *bridge-domain-name*],
[edit vlans *vlan-name*]

[edit chassis (QFX Series)]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.3 for QFX Series switches.
Hierarchy level **[edit vlans *vlan-name*]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Hierarchy level **[edit bridge-domains *bridge-domain-name*]** introduced in Junos OS Release 14.1 for MX Series routers.
custom-profile option introduced in Junos OS Release 15.1x53-D30 for QFX5200 Series switches only.

Description Configure a unified forwarding table profile to allocate the amount of memory available for the following:

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

This feature enables you to select a profile that optimizes the amount of memory available for various types of forwarding-table entries based on the needs of your network. For example, for a switch that handles a great deal of Layer 2 traffic, such as a virtualized network with many servers and virtualized machines, you would choose the **l2-profile-one**, which allocates the highest amount of memory to MAC addresses.

You configure the memory allocation for LPM table entries differently, depending on whether you using Junos OS Release 13.2X51-D10 or Junos OS Release 13.2X51-D15 and later. For more information about configuring memory allocation for LPM table entries, see "Configuring the Unified Forwarding Table on Switches" on page 37.

The **num-65-127-prefix** ***number*** statement is not supported on the **custom-profile** and the **lpm-profile**. The **prefix-65-127-disable** and **unicast-in-lpm** statements are supported only on the **lpm-profile**.

When you commit a configuration with a forwarding table profile change, in most cases the Packet Forwarding Engine restarts automatically to apply the new parameters, which brings the data interfaces down and then up again.

However, starting with Junos OS Releases 14.1X53-D40, 15.1R5, and 16.1R3, for a Virtual Chassis or Virtual Chassis Fabric (VCF) comprised of EX4600 or QFX5100 switches, the Packet Forwarding Engine in member switches does not automatically restart upon configuring and committing a unified forwarding table profile change. This behavior avoids having Virtual Chassis or VCF instability and a prolonged convergence period if a profile change is propagated to member switches and multiple Packet Forwarding Engines all restart at the same time. In this environment, instead of automatically restarting when you initially commit a profile configuration change, the message **Reboot required for configuration to take effect** is displayed at the master switch CLI prompt, notifying you that the profile change does not take effect until the next time you restart the Virtual Chassis or VCF. The profile configuration change is propagated to member switches that support this feature, and a reminder that a reboot is required to apply this pending configuration change appears in the system log of the master switch and applicable member switches. You then enable the profile change subsequently during a planned downtime period using the **request system reboot** command, which quickly establishes a stable Virtual Chassis or VCF with the new configuration.

NOTE: You should plan to make unified forwarding table profile changes only when you are ready to perform a Virtual Chassis or VCF system reboot *immediately* after committing the configuration update. Otherwise, in the intervening period between committing the configuration change and rebooting the Virtual Chassis or VCF, the system can become inconsistent if

a member experiences a problem and restarts. In that case, the new configuration takes effect on the member that was restarted, while the change is not yet activated on all the other members.

The remaining statements are explained separately. See CLI Explorer.

Options **profile-name**—name of the profile to use for memory allocation in the unified forwarding table. Table 121 on page 969 lists the profiles you can choose that have set values and the associated values for each type of entry.

On QFX5200 Series switches only, you can also select **custom-profile**. This profile enables you to allocate from one to four banks of shared hash memory to a specific type of forwarding-table entry. Each shared hash memory bank can store a maximum of the equivalent of 32,000 IPv4 unicast addresses.

Table 121: 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 Release13.2X51-D10. Starting in Junos OS Relase13.2X51-D15, the **lpm-profile** supports IPv4 and IPv6 entries.

NOTE: 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*.

l2-entries | l3-entries | lpm-entries—(**custom-profile only**) Select a type of forwarding-table entry—Layer 2, Layer 3, or LPM—to allocate a specific number of shared memory banks. You configure the amount of memory to allocate for each type of entry separately.

num-banks *number*—(**custom-profile only**) Specify the number of shared memory banks to allocate for a specific type of forwarding-table entry. Each shared memory bank stores the equivalent of 32,000 IPv4 unicast addresses.

Range: 0 through 4.

NOTE: There are four shared memory banks, which can be allocated flexibly among the three types of forwarding-table entries. To allocate

no shared memory for a particular entry type, specify the number 0. When you commit the configuration, the system issues a commit check to ensure that you have not configured more than four memory banks. You do not have to configure all four shared memory banks. By default, each entry type is allocated the equivalent of 32,000 IPv4 unicast addresses in shared memory.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding the Unified Forwarding Table*
- Example: Configuring a Unified Forwarding Table Custom Profile on QFX Series Switches on page 47
- *Configuring Traffic Forwarding and Monitoring*

global-mac-limit (Protocols)

Syntax

```
global-mac-limit limit {
    packet-action drop;
}
```

Hierarchy Level [edit protocols l2-learning]

Release Information Statement modified in Junos OS Release 9.5.

Description Limit the number of media access control (MAC) addresses learned from the logical interfaces on the router.

Default 131,071 MAC addresses

NOTE: SRX300, SRX320, SRX340, and SRX345 devices support 16,383 addresses, and SRX1500 devices support 24,575 addresses.

Options *limit*—Number of MAC addresses that can be learned on the device.

Range: 20 through 13,1071 addresses

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring VLANs on Security Devices on page 382

global-mac-move

Syntax

```
global-mac-move {
  cooloff-time seconds;
  disable-action;
  exclusive-mac virtual-mac-mac-address/mask;
  interface-recovery-time seconds;
  notification-time seconds;
  reopen-time seconds;
  statistical-approach-wait-time seconds;
  threshold-count count;
  threshold-time seconds;
  virtual-mac mac-address /mask;
}
```

Hierarchy Level [edit protocols l2-learning]

Release Information

Statement introduced in Junos OS Release 9.4.
Support for logical systems added in Junos OS Release 9.6.
Support for disable-action and reopen-time added in Junos OS Release 13.2.
Support for exclusive-mac added in Junos OS Release 14.1X53-D45.
Statements **cooloff-time**, **interface-recovery-time**, **statistical-approach-wait-time**, and **virtual-mac** moved from vpls-mac-move to global-mac-move hierarchy level in Junos OS Release 17.4.

Description Set parameters for media access control (MAC) address move reporting.

Default By default, MAC moves notify every second, with a threshold time of 1 second and a threshold count of 50.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Configuring MAC Move Parameters on page 54
- *MAC Moves Loop Prevention in VPLS Network Overview*
- *Example: Configuring Loop Prevention in VPLS Network Due to MAC Moves*
- *virtual-mac*

global-mac-statistics

Syntax global-mac-statistics;

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in Junos OS Release 9.2.
Support for logical systems added in Junos OS Release 9.6.

Description (MX Series routers and EX Series switches only) Enable MAC accounting for the entire router or switch.

Default disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Enabling MAC Accounting*

global-mac-table-aging-time

Syntax global-mac-table-aging-time *seconds*;

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in Junos OS Release 9.2.
Statement modified in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.

Description Configure the timeout interval for entries in the MAC table.

Default 300 seconds

Options ***seconds***—Time elapsed before MAC table entries are timed out and entries are deleted from the table.

Range: For MX Series routers: 10 through 1 million; for EX Series and QFX Series switches: 60 through 1 million; for SRX devices: 10 through 64,000 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring the MAC Table Timeout Interval*
- Configuring MAC Table Aging on Switches on page 81
- Example: Configuring VLANs on Security Devices on page 382

global-mode (Protocols)

Syntax global-mode (switching | transparent-bridge) ;

Hierarchy Level [edit protocols l2-learning]

Release Information Statement introduced in Junos OS Release 15.1X49-D40.

Description Specify the global mode for the SRX Series device as Layer 2 transparent bridge mode or switching mode. After changing the mode, you must reboot the device for the configuration to take effect.

Default On SRX1500, the default Layer 2 global mode is transparent-bridge mode.

Starting with Junos OS Release 15.1X49-D100, on SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M devices, the default Layer 2 global mode configuration is changed from transparent-bridge to switching mode.

NOTE: You must explicitly configure Layer 2 transparent-bridge mode for the SRX300, SRX320, SRX340, SRX345, SRX550, and SRX550M devices that work in transparent mode. Use the command **set protocols l2-learning global-mode transparent-bridge** before rebooting the devices with Junos OS 15.1X49-D100 image.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- l2-learning on page 1012
- Ethernet Switching and Layer 2 Transparent Mode Overview on page 25

global-no-mac-learning

Syntax global-no-mac-learning;

Hierarchy Level [edit protocols l2-learning],
[edit protocols l2-learning]

Release Information Statement introduced in Junos OS Release 9.2.
Statement modified for SRX Series in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.

Description Disable MAC learning on the entire device.

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

- Disabling Layer 2 Learning and Forwarding on page 200
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Example: Configuring VLANs on Security Devices on page 382

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. See CLI Explorer.

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*

group (Redundant Trunk Groups)

Syntax

```
group name {
  interface interface-name <primary>;
    interface interface-name;
  preempt-cutover-timer seconds;
}
```

Hierarchy Level

- For platforms with ELS:

  [edit switch-options redundant-trunk-group]

- For platforms without ELS:

  [edit ethernet-switching-options redundant-trunk-group]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 3 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Create a redundant trunk group.

Options

name—The name of the redundant trunk group.

- For platforms with ELS:

  The group name must be a string "rtg*n*" where *n* is a number from 0 through 15, such as "rtg2" or "rtg10".

- For platforms without ELS:

  The group name must start with a letter and can consist of letters, numbers, dashes, and underscores.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support on page 613
- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

guard-interval

Syntax guard-interval *number*;

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description When a link goes down, the ring protection link (RPL) activates. When the downed link comes back up, the RPL link receives notification, restores the link, and waits for the restore interval before issuing another block on the same link. This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value.

Options *number*—Guard timer interval, in milliseconds.

Range: 10 through 2000 ms

Default: 500 ms

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

hold-interval (Protection Group)

Syntax hold-interval *number*;

Hierarchy Level [edit protocols protection-group ethernet-ring *name*]

Release Information Statement introduced in Junos OS Release 9.4.

Description Specify the hold-off timer interval *for all rings* in 100 millisecond (ms) increments.

Options ***number***—Hold-timer interval, in milliseconds.
Range: 0 through 10,000 ms
Default: 100 ms

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435

host-inbound-traffic

Syntax

```
host-inbound-traffic {
  protocols protocol-name {
    except;
  }
  system-services service-name {
    except;
  }
}
```

Hierarchy Level

[edit security zones functional-zone management],
[edit security zones functional-zone management interfaces *interface-name*],
[edit security zones security-zone *zone-name*],
[edit security zones security-zone *zone-name* interfaces *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Control the type of traffic that can reach the device from interfaces bound to the zone.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Understanding How to Control Inbound Traffic Based on Traffic Types*
- *Understanding How to Control Inbound Traffic Based on Protocols*

inet6 (Security Forwarding Options)

Syntax

```
inet6 {
  mode (drop | flow-based | packet-based);
}
```

Hierarchy Level [edit security forwarding-options family]

Release Information Statement introduced in Junos OS Release 8.5.

Description Enable packet-based or flow-based processing of IPv6 traffic. By default, the device drops IPv6 traffic.

NOTE: Packet-based processing is not supported on the following SRX Series devices: SRX1500, SRX5600, and SRX5800.

Options The **mode** statement is described separately.

Required Privilege Level
security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- family inet6 on page 955

inner-tag-protocol-id

Syntax inner-tag-protocol-id *tpid*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Configure the IEEE 802.1Q TPID value to rewrite for the inner tag.

All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options** ethernet-switch-profile tag-protocol-id **[*tpids*]]** hierarchy level.

On MX Series routers, you can use this statement for Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, and for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs.

Default If the **inner-tag-protocol-id** statement is not configured, the TPID value is 0x8100.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Inner and Outer TPIDs and VLAN IDs*

inner-vlan-id

Syntax inner-vlan-id *number*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description For Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, and for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers or 100-Gigabit Ethernet Type 5 PIC with CFP, or on Ethernet interfaces on EX Series switches, specify the VLAN ID to rewrite for the inner tag of the final packet.

You cannot include the **inner-vlan-id** statement with the **swap** statement, **swap-push** statement, **push-push** statement, or **push-swap** statement and the **inner-vlan-id** statement at the **[edit interfaces *interface-name* unit *logical-unit-number*** output-vlan-map**]** hierarchy level. If you include any of those statements in the output VLAN map, the VLAN ID in the outgoing frame is rewritten to the **inner-vlan-id** statement you include at the **[edit interfaces *interface-name* unit *logical-unit-number*]** hierarchy level.

Options *number*—VLAN ID number.

Range: 0 through 4094

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Inner and Outer TPIDs and VLAN IDs*

input-vlan-map

Syntax

```
input-vlan-map {
  (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
  inner-tag-protocol-id tpid;
  inner-vlan-id number;
  tag-protocol-id tpid;
  vlan-id number;
}
```

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.
pop-pop, **pop-swap**, **push-push**, **swap-push**, and **swap-swap** statements introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X50-D15 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

For Gigabit Ethernet IQ, 10-Gigabit Ethernet SFPP interfaces, 100-Gigabit Ethernet Type 5 PIC with CFP only as well as Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces, define the rewrite profile to be applied to incoming frames on this logical interface.

The remaining statements are explained separately. See CLI Explorer.

NOTE: Connectivity fault management (CFM) sessions for all interfaces in which **input-vlan-map** is configured are supported only if the interface also has an explicit configuration for **output-vlan-map** as **output-vlan-map pop;**. See output-vlan-map. This configuration is required for all the interfaces in the topology even when the CFM session is on that interface or on a different interface in the data path of the same topology.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Stacking a VLAN Tag*
- output-vlan-map on page 1058
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

instance-type

Syntax instance-type *type*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*],
[edit routing-instances *routing-instance-name*]

Release Information Statement introduced before Junos OS Release 7.4.
virtual-switch and **layer2-control** options introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
mpls-internet-multicast option introduced in Junos OS Release 11.1 for the EX Series, M Series, MX Series, and T Series.
evpn option introduced in Junos OS Release 13.2 for MX 3D Series routers.
evpn option introduced in Junos OS Release 17.3 for the QFX Series.
forwarding option introduced in Junos OS Release 14.2 for the PTX Series.
mpls-forwarding option introduced in Junos OS Release 16.1 for the MX Series.
evpn-vpws option introduced in Junos OS Release 17.1 for MX Series routers.
Support for logical systems on MX Series routers added in Junos OS Release 17.4R1.

Description Define the type of routing instance.

Options

NOTE: On ACX Series routers, you can configure only the forwarding, virtual router, and VRF routing instances.

type— Can be one of the following:

- **evpn**—(MX 3D Series routers, QFX switches and EX9200 switches)— Enable an Ethernet VPN (EVPN) on the routing instance.
  hierarchy level.
- **evpn-vpws**—Enable an Ethernet VPN (EVPN) Virtual Private Wire Service (VPWS) on the routing instance.
- **forwarding**—Provide support for filter-based forwarding, where interfaces are not associated with instances. All interfaces belong to the default instance. Other instances are used for populating RPD learned routes. For this instance type, there is no one-to-one mapping between an interface and a routing instance. All interfaces belong to the default instance inet.0.
- **l2backhaul-vpn**—Provide support for Layer 2 wholesale VLAN packets with no existing corresponding logical interface. When using this instance, the router learns both the outer tag and inner tag of the incoming packets, when the **instance-role** statement is defined as **access**, or the outer VLAN tag only, when the **instance-role** statement is defined as **nni**.

- **l2vpn**—Enable a Layer 2 VPN on the routing instance. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- layer2-control— (MX Series routers only) Provide support for RSTP or MSTP in customer edge interfaces of a VPLS routing instance. This instance type cannot be used if the customer edge interface is multihomed to two provider edge interfaces. If the customer edge interface is multihomed to two provider edge interfaces, use the default BPDU tunneling.
- **mpls-forwarding**—(MX Series routers only) Allow filtering and translation of route distinguisher (RD) values in IPv4 and IPv6 VPN address families on both routes received and routes sent for selected BGP sessions. In particular, for Inter-AS VPN Option-B networks, this option can prevent the malicious injection of VPN labels from one peer AS boundary router to another.
- **mpls-internet-multicast**—(EX Series, M Series, MX Series, and T Series routers only) Provide support for ingress replication provider tunnels to carry IP multicast data between routers through an MPLS cloud, using MBGP or next-generation MVPN.
- **no-forwarding**—This is the default routing instance. Do not create a corresponding forwarding instance. Use this routing instance type when a separation of routing table information is required. There is no corresponding forwarding table. All routes are installed into the default forwarding table. IS-IS instances are strictly nonforwarding instance types.
- **virtual-router**—Enable a virtual router routing instance. This instance type is similar to a VPN routing and forwarding instance type, but used for non-VPN-related applications. You must configure the **interface** statement for this type of routing instance. You do not need to configure the **route-distinguisher**, **vrf-import**, and **vrf-export** statements.
- **virtual-switch**—(MX Series routers, EX9200 switches, and QFX switches only) Provide support for Layer 2 bridging. Use this routing instance type to isolate a LAN segment with its Spanning Tree Protocol (STP) instance and to separate its VLAN identifier space.
- **vpls**—Enable VPLS on the routing instance. Use this routing instance type for point-to-multipoint LAN implementations between a set of sites in a VPN. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.
- **vrf**—VPN routing and forwarding (VRF) instance. Provides support for Layer 3 VPNs, where interface routes for each instance go into the corresponding forwarding table only. Required to create a Layer 3 VPN. Create a VRF table (***instance-name*.inet.0**) that contains the routes originating from and destined for a particular Layer 3 VPN. For this instance type, there is a one-to-one mapping between an interface and a routing instance. Each VRF instance corresponds with a forwarding table. Routes on an interface go into the corresponding forwarding table. You must configure the **interface**, **route-distinguisher**, **vrf-import**, and **vrf-export** statements for this type of routing instance.

Required Privilege Level
routing— To view this statement in the configuration.
routing-control— To add this statement to the configuration.

Related Documentation
- *Configuring the Instance Type*
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- *Configuring Virtual Router Routing Instances*
- *Example: Configuring Filter-Based Forwarding on the Source Address*
- *Example: Configuring Filter-Based Forwarding on Logical Systems*

inter-switch-link

Syntax
inter-switch-link vlan members *primary-vlan-name*;

Hierarchy Level
[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching interface-mode trunk]

Release Information
Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.

Description
Use this configuration statement when a private VLAN (PVLAN) spans multiple switches. The Inter-Switch Link protocol (ISL) must be configured on a trunk port of the primary VLAN in order to connect the switches composing the PVLAN to each other. You do not need to configure an ISL when a PVLAN is configured on a single switch.

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 Spanning Multiple EX Series Switches (CLI Procedure) on page 277

interface

Syntax interface *interface-name*;

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 8.4.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2.
In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 12.3X52 for ACX Series routers.
Statement introduced in Junos OS Release 15.1.

Description (MX Series routers and EX Series switches only) Specify the logical interfaces to include in the bridge domain, VLAN, VPLS instance, or virtual switch.

Options ***interface-name***—Name of a logical interface. For more information about how to configure logical interfaces, see the *Junos OS Network Interfaces Library for Routing Devices*.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring a Bridge Domain*
- *Configuring a Layer 2 Virtual Switch*
- Configuring a Layer 2 Virtual Switch on an EX Series Switch on page 95
- *Tunnel Services Overview*
- *Tunnel Interface Configuration on MX Series Routers Overview*

interface (MVRP)

Syntax

```
interface (all | interface-name) {
  disable;
  join-timer milliseconds;
  leave-timer milliseconds;
  leaveall-timer milliseconds;
  registration (forbidden | normal);
}
```

Syntax

```
interface (all | interface-name) {
  join-timer milliseconds;
  leave-timer milliseconds;
  leaveall-timer milliseconds;
  point-to-point;
  registration (forbidden | normal | restricted);
}
```

Hierarchy Level

[edit protocols mvrp]

[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type)

[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

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. See CLI Explorer.

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 on page 535
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Verifying That MVRP Is Working Correctly on page 551

interface (Layer 2 Protocol Tunneling)

Syntax

```
interface interface-name {
  enable-all-ifl;
  protocol protocol-name;
}
```

Hierarchy Level

[edit logical-systems *name* protocols layer2-control mac-rewrite],
[edit protocols layer2-control mac-rewrite]

Release Information

Statement introduced in Junos OS Release 9.1.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
enable-all-if statement added in Junos OS Release 13.3.
Support for PVSTP protocol introduced in Junos OS Release 13.3.
Statement introduced in Junos OS Release 14.1X53-D10 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.

Description

Configure an interface for Layer 2 protocol tunneling.

NOTE: The enable-all-ifl option is available on EX9200 switches but not on other EX Series switches.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Layer 2 Protocol Tunneling Through a Network*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389

interface (Redundant Trunk Groups)

Syntax

interface *interface-name* <primary>;
interface *interface-name*;

Hierarchy Level

For platforms with ELS:

[edit switch-options redundant-trunk-group group *name*]

For platforms without ELS:

[edit ethernet-switching-options redundant-trunk-group group *name*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 3 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Configure a primary link and secondary link on trunk ports. If the primary link fails, the secondary link automatically takes over as the primary link without waiting for normal STP convergence.

Options

interface *interface-name*—A logical interface or an aggregated interface containing multiple ports.

primary—(Optional) Specify one of the interfaces in the redundant group as the primary link. The interface without this option is the secondary link in the redundant group. If a link is not specified as **primary**, the software compares the two links and selects the link with the highest port number as the active link. For example, if the two interfaces are **ge-0/1/0** and **ge-0/1/1**, the software assigns **ge-0/1/1** as the active link.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support on page 613
- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

interface (Routing Instances)

Syntax

```
interface interface-name {
  description text;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*],
[edit routing-instances *routing-instance-name*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 13.2 for MX 3D Series routers.

Description

Specify the interface over which the VPN traffic travels between the PE device and CE device. You configure the interface on the PE device. If the value **vrf** is specified for the **instance-type** statement included in the routing instance configuration, this statement is required.

Options

interface-name—Name of the interface.

The remaining statement is explained separately. Search for a statement in CLI Explorer or click a linked statement in the Syntax section for details.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Interfaces for VPN Routing*
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- *interface (VPLS Routing Instances)*

interface (Switching Options)

Syntax

```
interface interface-name {
  encapsulation-type;
  ignore-encapsulation-mismatch;
  pseudowire-status-tlv;
  static-mac mac-address {
    vlan-id vlan-id;
  }
}
```

Hierarchy Level [edit vlans *vlans-name* switch-options]

Release Information Statement modified in Junos OS Release 9.5.

Description Specify the logical interfaces to include in the VLAN.

Options

- ***interface-name***—Name of a logical interface.
- ***encapsulation-type***—Encapsulation type for VPN.
- ***ignore-encapsulation-mismatch***—Allow different encapsulation types on local and remote devices.
- ***pseudowire-status-tlv***—Send pseudowire status.
- ***mac-address***—Static MAC address assigned to the logical interface.
- ***vlan-id***—VLAN identifier.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding VLANs on Security Devices on page 380

interface (VLANs)

List of Syntax

Syntax (QFX Series, QFabric, NFX Series and EX4600) on page 994
Syntax (EX Series and SRX210) on page 994

Syntax (QFX Series, QFabric, NFX Series and EX4600)

```
interface interface-name {
  mapping (native (push | swap) | tag (push | swap));
}
```

Syntax (EX Series and SRX210)

```
interface interface-name {
  egress;
  ingress;
  mapping (native (push | swap) | policy | tag (push | swap));
  pvlan-trunk;
}
```

QFX Series, QFabric, NFX Series and EX4600

[edit vlans *vlan-name*]

EX Series and SRX210

[edit vlans *vlan-name*],
[edit vlans *vlan-name*],
[edit vlans *vlan-name* vlan-id *number*],
[edit vlans *vlan-name* vlan-id *number*],
[edit vlans *vlan-name* vlan-id-list *number*]

Release Information

Statement introduced in Junos OS Release 9.3 for EX Series switches.
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 Ethernet interface

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.system-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Configuring VLANs on Switches on page 93
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98
- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

interface-mac-limit

Syntax

```
interface-mac-limit {
  limit
  disable;
  packet-action ;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* bridge-options],
[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface *interface-name*],
[edit logical-systems *logical-system-name* switch-options],
[edit logical-systems *logical-system-name* switch-options interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* switch-options],
[edit routing-instances *routing-instance-name* switch-options interface *interface-name*],
[edit switch-options],
[edit switch-options],
[edit switch-options interface *interface-name*],
[edit switch-options interface *interface-name*],
[edit vlans *vlan-name* switch-options],
[edit vlans *vlan-name* switch-options interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options], [edit switch-options interface *interface-name*], [edit vlans *vlan-name* switch-options], and [edit vlans *vlan-name* switch-options interface *interface-name*] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Configure a limit to the number of MAC addresses that can be learned from a bridge domain, VLAN, virtual switch, or set of bridge domains or VLANs.

NOTE: For multichassis link aggregation (MC-LAG) peers in active-active mode, configuring the **interface-mac-limit** statement or changing the **interface-mac-limit** configuration when traffic is flowing can cause the MAC entries to be out of synchronization between the two MC-LAG peers, which might result in flooding. To avoid flooding, you must either halt traffic forwarding and then configure the **interface-mac-limit** statement or use the **commit at** configuration statement to commit the changes at the same time in both the peer nodes.

Alternatively, if flooding does occur, you can clear the bridge MAC table on both the routers or switches by using the **clear bridge mac-table** command. Running this command ensures that the MAC entries are re-learned and in synchronization between both the peers.

Default The default MAC limit varies with the platform.

Options

disable—Disables the global interface-mac-limit configuration on an interface and sets the maximum interface-mac-limit that is permitted on the device.

limit—Sets the maximum number of MAC addresses learned from an interface.

Range: 1 through <default MAC limit> MAC addresses per interface. Range is platform specific.

If you configure both **disable** and **limit**, disable takes precedence and packet-action is set to **none**. 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

- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port on page 200

interface-mode

Syntax interface-mode (access | trunk <inter-switch-link>);

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family bridge],
[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* family bridge]

Release Information Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 15.1.
inter-switch-link option introduced in Junos OS Release 14.2 for MX240, MX480, and MX960 routers in enhanced LAN mode.

Description

NOTE: This statement supports the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see port-mode. For ELS details, see "Using the Enhanced Layer 2 Software CLI" on page 3.

QFX3500 and QFX3600 standalone switches—Determine whether the logical interface accepts or discards packets based on VLAN tags. Specify the **trunk** option to accept packets with a VLAN ID that matches the list of VLAN IDs specified in the **vlan-id** or **vlan-id-list** statement, then forward the packet within the bridge domain or VLAN configured with the matching VLAN ID. Specify the **access** option to accept packets with no VLAN ID, then forward the packet within the bridge domain or VLAN configured with the VLAN ID that matches the VLAN ID specified in the **vlan-id** statement.

NOTE: On MX Series routers, if you want IGMP snooping to be functional for a bridge domain, then you should not configure interface-mode and irb for that bridge. Such a configuration commit succeeds, but IGMP snooping is not functional, and a message informing the same is displayed. For more information, see *Configuring a Trunk Interface on a Bridge Network*.

Options

access—Configure a logical interface to accept untagged packets. Specify the VLAN to which this interface belongs using the **vlan-id** statement.

trunk—Configure a single logical interface to accept packets tagged with any VLAN ID specified with the **vlan-id** or **vlan-id-list** statement.

trunk inter-switch-link—For a private VLAN, configure the InterSwitch Link protocol (ISL) on a trunk port of the primary VLAN in order to connect the switches composing the

PVLAN to each other. You do not need to configure an ISL when a PVLAN is configured on a single switch. This configuration specifies whether the particular interface assumes the role of interswitch link for the PVLAN domains of which it is a member. This option is supported only on MX240, MX480, and MX960 routers in enhanced LAN mode.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Access Mode on a Logical Interface*
- *Configuring a Logical Interface for Trunk Mode*
- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support on page 170
- *Tunnel Services Overview*
- *Tunnel Interface Configuration on MX Series Routers Overview*

interfaces (CoS)

Syntax

```
interfaces
  interface-name {
    input-scheduler-map map-name ;
    input-shaping-rate rate ;
    scheduler-map map-name ;
    scheduler-map-chassis map-name ;
    shaping-rate rate ;
    unit logical-unit-number {
      adaptive-shaper adaptive-shaper-name ;
      classifiers {
        (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence)
        ( classifier-name | default);
      }
      forwarding-class class-name ;
      fragmentation-map map-name ;
      input-scheduler-map map-name ;
      input-shaping-rate (percent percentage | rate );
      input-traffic-control-profile profiler-name shared-instance instance-name ;
      loss-priority-maps {
        default;
        map-name ;
      }
      output-traffic-control-profile profile-name shared-instance instance-name ;
      rewrite-rules {
        dscp ( rewrite-name | default);
        dscp-ipv6 ( rewrite-name | default);
        exp ( rewrite-name | default) protocol protocol-types ;
        frame-relay-de ( rewrite-name | default);
        inet-precedence ( rewrite-name | default);
      }
      scheduler-map map-name ;
      shaping-rate rate ;
      virtual-channel-group group-name ;
    }
  }
}
```

Hierarchy Level [edit class-of-service interface *interface-name* unit *number*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Associate the class-of-service configuration elements with an interface.

Options interface *interface-name* unit *number*—The user-specified interface name and unit number.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *Class of Service Feature Guide for Security Devices*

interfaces (Q-in-Q Tunneling)

Syntax
```
interfaces interface-name {
  no-mac-learning;
}
```

Hierarchy Level [edit ethernet-switching-options]

Release Information Statement introduced in Junos OS Release 9.5 for EX Series switches.

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. See CLI Explorer.

Required Privilege Level system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

Related Documentation
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554

interfaces (Security Zones)

Syntax

```
interfaces interface-name {
  host-inbound-traffic {
    protocols protocol-name {
      except;
    }
    system-services service-name {
      except;
    }
  }
}
```

Hierarchy Level [edit security zones functional-zone management],
[edit security zones security-zone *zone-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the set of interfaces that are part of the zone.

Options ***interface-name*** —Name of the interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Understanding Security Zones*

interfaces

List of Syntax

Syntax (QFX Series) on page 1002
Syntax (EX Series, MX Series and T Series) on page 1002

Syntax (QFX Series)

```
interfaces interface-name {
  no-mac-learning;
}
```

Syntax (EX Series, MX Series and T Series)

```
interfaces { ... }
```

QFX Series

[edit ethernet-switching-options]

EX Series, MX Series and T Series

[edit]

Release Information

Statement introduced before Junos OS Release 7.4.
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**.

Default

The management and internal Ethernet interfaces are automatically configured. You must configure all other interfaces.

Options

interface-name —Name of an interface that is configured for **family ethernet-switching**.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Physical Interface Configuration Statements Overview*
- *Configuring Aggregated Ethernet Link Protection*

irb (Interfaces)

Syntax

```
irb {
  accounting-profile name;
  arp-l2-validate;
  description  text;

  (gratuitous-arp-reply | no-gratuitous-arp-reply);
  hold-time up milliseconds down milliseconds;
  mtu bytes;
  no-gratuitous-arp-request;

  traceoptions {
    flag flag;
  }
  (traps | no-traps);
  unit logical-unit-number {
    accounting-profile name;
    bandwidth rate;
    description text;
    enhanced-convergence;
    disable;
    encapsulation type;
    family inet {
      accounting {
        destination-class-usage;
        source-class-usage {
          input;
          output;
        }
      }
      address ipv4-address {
        arp ip-address (mac | multicast-mac) mac-address <publish>;
        broadcast address;
        preferred;
        primary;
        vrrp-group group-number {
          (accept-data | no-accept-data);
          advertise–interval seconds;
          advertisements-threshold number;
          authentication-key key;
          authentication-type authentication;
          fast-interval milliseconds;
          (preempt | no-preempt) {
            hold-time seconds;
          }
          priority number;
          track {
            interface interface-name {
              bandwidth-threshold bandwidth;
              priority-cost number;
            }
            priority-hold-time seconds;
            route ip-address/mask routing-instance instance-name priority-cost cost;
```

```
            }
            virtual-address [ addresses ];
            vrrp-inherit-from {
              active-group group-number;
              active-interface interface-name;
            }
          }
        }
        filter {
          input filter-name;
          output filter-name;
        }
        mtu bytes;
        no-neighbor-learn;
        no-redirects;
        primary;
        rpf-check {
          fail-filter filter-name;
          mode {
            loose;
          }
        }
        targeted-broadcast {
          forward-and-send-to-re;
          forward-only;
        }
      }
      family inet6 {
        accounting {
          destination-class-usage;
          source-class-usage {
            input;
            output;
          }
        }
        address address {
          eui-64;
          ndp ip-address (mac | multicast-mac) mac-address <publish>;
          preferred;
          primary;
          vrrp-inet6-group group-id {
            accept-data | no-accept-data;
            advertisements-threshold number;
            authentication-key key;
            authentication-type authentication;
            fast-interval milliseconds;
            inet6-advertise-interval milliseconds;
            preempt | no-preempt {
              hold-time seconds;
            }
            priority number;
            track {
              interface interface-name {
                bandwidth-threshold bandwidth priority-cost number;
                priority-cost number;
              }
```

```
                priority-hold-time seconds;
                route ip-address/mask routing-instance instance-name priority-cost cost;
              }
              virtual-inet6-address [addresses];
              virtual-link-local-address ipv6–address;
              vrrp-inherit-from {
                active-group group-number;
                active-interface interface-name;
              }
            }
          }
          (dad-disable | no-dad-disable);
          filter {
            input filter-name;
            output filter-name;
          }
          mtu bytes;
          nd6-stale-time seconds;
          no-neighbor-learn;
          no-redirects;
          policer {
            input policer-name;
            output policer-name;
          }
          rpf-check {
            fail-filter filter-name;
            mode {
              loose;
            }
          }
        }
        family iso {
          address interface-address;
          mtu bytes;
        }
        family mpls {
          filter {
            input filter-name;
            output filter-name;
          }
          mtu bytes;
          policer {
            input policer-name;
            output policer-name;
          }
        }
        native-inner-vlan-id vlan-id;
        proxy-arp (restricted | unrestricted);
        (traps | no-traps);
        vlan-id-list [vlan-id's];
        vlan-id-range [vlan-id-range];
      }
    }
```

Hierarchy Level [edit interfaces *interface-name*

Release Information Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
irb option introduced in Junos OS Release 13.2 for the QFX Series.

Description Configure the properties of a specific integrated bridging and routing (IRB) interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

isid

Syntax

```
isid isid-number vlan-id-list [ vlan-ids ] {
  source-bmac <mac-address> <length>
}
```

Hierarchy Level [edit routing-instances *routing-instance-name* service-groups *service-group-name*]

Release Information Statement introduced in JUNOS Release 10.0.

Description For IEEE 802.1ah provider backbone bridge (PBB) configurations, configure the service identifier (I-SID) for the customer routing instance (I-component) service group.

Options ***isid***—Service identifier. Enter an I-SID in the range from **256** through **16777214**.

vlan-id-list [*vlan-ids*]—List of service VLANs (S-VLANs).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

isid-list

Syntax isid-list all-service-groups;

Hierarchy Level [edit interfaces *pseudo-logical-interface-name* unit *logical-unit-number* family bridge]

Release Information Statement introduced in JUNOS Release 10.0.

Description For IEEE 802.1ah provider backbone bridge (PBB) configurations, map all service identifiers (I-SIDs) specified for the service groups.

Options **all-service-groups**—Map all service identifiers (I-SIDs) for the specified service groups.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

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 QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- Understanding Secondary VLAN Trunk Ports and Promiscuous Access Ports on PVLANs on page 244
- Example: Configuring PVLANs with Secondary VLAN Trunk Ports and Promiscuous Access Ports on a QFX Series Switch on page 342

isolated-vlan

Syntax isolated-vlan vlan-name *isolated-vlan-name* vlan-id *isolated-vlan-id*;

Hierarchy Level [edit vlans *primary-vlan-name* vlan-id *primary-vlan-vlan-id*]

Release Information Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.

Description Configure the specified isolated VLAN to be a secondary VLAN of the specified primary VLAN. An isolated VLAN receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN.

NOTE: Before you specify this configuration statement, you must have already configured an isolated VLAN and assigned a VLAN ID to it. See private-vlan.

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 with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277

isolation-id

Syntax isolation-id *number*;

Hierarchy Level [edit vlans *vlan-name* vlan-id *number*]

Release Information Statement introduced in Junos OS Release 10.4 for EX Series switches.

Description Configure an inter-switch isolated VLAN within a private VLAN (PVLAN) 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 Spanning Multiple EX Series Switches (CLI Procedure)*

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 QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275

join-timer (MVRP)

Syntax join-timer *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type),
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit logical-systems *logical-system-name* protocols mvrp interface (all | *interface-name*)],

[edit protocols mvrp interface (all | *interface-name*)]

[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type)

Release Information
Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

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***—Interval that the interface must wait before sending MVRP PDUs (range from 100 milliseconds through 500 milliseconds). 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.

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 1020
- leaveall-timer on page 1022
- point-to-point (MVRP) on page 1065
- registration on page 1095
- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Verifying That MVRP Is Working Correctly on page 551

l2-learning

List of Syntax

Syntax (MX Series, QFX Series, EX Series) on page 1012
Syntax (SRX Series) on page 1012

Syntax (MX Series, QFX Series, EX Series)

```
l2-learning {
  global-le-bridge-domain-aging-time;
  global-mac-ip-limit number;
  global-mac-ip-table-aging-time seconds;
  global-mac-limit limit;
  global-mac-statistics;
  global-mac-table-aging-time seconds;
  global-no-mac-learning;
  global-mac-move;
}
```

Syntax (SRX Series)

```
l2-learning {
  global-mac-limit limit {
  packet-action-drop
  }
  global-mac-table-aging-time seconds;
  global-mode (switching | transparent-bridge) ;
  global-no-mac-learning;
}
```

Hierarchy Level

[edit protocols]

Release Information

Statement introduced in Junos OS Release 8.4.
Statement modified in Junos OS Release 9.5. Support for global mode added in Junos OS Release 15.1X49-D40.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series.
global-le-bridge-domain-aging-time option introduced in Junos OS Release 14.2R5 for the MX Series.
global-mac-ip-limit and **global-mac-ip-table-aging-time** options introduced in Junos OS Release 17.4R1 for MX Series routers and EX9200 switches.

Description

Configure Layer 2 address learning and forwarding properties globally.

The remaining statements are explained separately. See CLI Explorer.

Options

global-le-bridge-domain-aging-time—Specify the aging time of LE bridge-domain. The MAC address is learnt after next hop(NH) and bridge-domain(BD), also called NHBD. This aging time delays the deletion of NHBD. Configuring lesser time, in seconds, results in faster deletion of NHBD.

Range: 120 to 1000000 seconds

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Understanding Layer 2 Learning and Forwarding*
- global-mac-table-aging-time on page 974
- global-mac-limit (Protocols) on page 971
- global-no-mac-learning on page 976
- global-mode (Protocols) on page 975

l3-interface (VLAN)

Syntax

l3-interface (vlan.*logical-interface-number* | irb.*logical-interface-number*);

l3-interface *l3-interface-name.logical-interface-number* {
  l3-interface-ingress-counting;
}

l3-interface *interface-name-logical-unit-number*;

Hierarchy Level

[edit vlans *vlan-name*]

[edit interfaces ge-*chassis/slot/port* unit *logical-unit-number* family ethernet-switching]
[edit vlans *vlan-name*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
[edit vlans *vlan-name*] hierarchy level introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.
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. Within a VLAN, traffic is bridged, while across VLANs, traffic is routed. Because traffic between VLANs must be routed, a common Layer 3 interface is required.

Default

No Layer 3 (routing) interface is associated with the VLAN.

Options

interface-name–logical-unit-number—Name of a logical interface.

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 (QFX Series) 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 (QFX Series) 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
- Configuring Routed VLAN Interfaces on Switches (CLI Procedure) on page 365
- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) on page 456
- show ethernet-switching interfaces on page 1215
- show ethernet-switching interface on page 1212
- show vlans on page 1510

l3-interface-ingress-counting

Syntax
l3-interface-ingress-counting *layer-3-interface-name*;

Hierarchy Level
[edit vlans *vlan-name*]

Release Information
Statement introduced in Junos OS Release 11.3 for EX Series switches.

Description
(EX8200 standalone switch and EX8200 Virtual Chassis) Enable routed VLAN interface (RVI) input counters on an EX8200 switch to collect RVI source statistics for tracking or billing purposes. The input counter is maintained by a firewall filter. The switch can maintain a limited number of firewall filter counters—these counters are allocated on a first-come, first-served basis.

Output (egress) counters for EX8200 switches are always present and cannot be removed.

Reset ingress-counting statistics with the *clear interfaces statistics* command.

Default
The input (ingress) counters (both packets and bytes) are disabled on an RVI by default.

Required Privilege Level
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation
- show vlans on page 1510
- *clear interfaces statistics*
- *Configuring Firewall Filters (CLI Procedure)*
- *firewall*
- Configuring Routed VLAN Interfaces on Switches (CLI Procedure) on page 365

layer2-control

Syntax

```
layer2-control {
  bpdu-block {
    disable-timeout seconds;
    interface interface-name;
  }
  mac-rewrite {
    interface interface-name {
      enable-all-ifl;
      protocol protocol-name;
    }
  }
  nonstop-bridging;
  traceoptions {
    file filename <files number> <size maximum-file-size> <world-readable |
      no-world-readable>;
    flag flag <disable>;
  }
}
```

Hierarchy Level [edit protocols]

Release Information

Statement introduced in Junos OS Release 8.4.
bpdu-block statement added in Junos OS Release 9.4.
enable-all-if statement added in Junos OS Release 13.3.
Statement introduced in Junos OS Release 14.1X53-D10 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D50 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.

Description Configure Layer 2 control protocols to enable features such as Layer 2 protocol tunneling or nonstop bridging.

The remaining statements are explained separately. See CLI Explorer.

NOTE: For a detailed description of configuring the **nonstop-bridging** statement, see the *Junos OS High Availability Library for Routing Devices*. When this statement is configured on routing platforms with two Routing Engines, a master Routing Engine switches over gracefully to a backup Routing Engine and preserves Layer 2 Control Protocol (L2CP) information.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Layer 2 Protocol Tunneling Through a Network*
- *Layer 2 Protocol Tunnel Configuration Guidelines*

- *Configuring Layer 2 Protocol Tunneling*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398
- **instance-type on page 985**

layer2-protocol-tunneling

Syntax

```
layer2-protocol-tunneling all | protocol-name {
  drop-threshold number;
  shutdown-threshold number;
}
```

Hierarchy Level [edit vlans *vlan-name* dot1q-tunneling]

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.

Description Enable Layer 2 protocol tunneling (L2PT) on the VLAN.

The remaining statements are explained separately. See CLI Explorer.

Default L2PT is not enabled.

Options **all**—Enable all supported Layer 2 protocols.

protocol-name—Name of the Layer 2 protocol. Values are:

- **802.1x**—IEEE 802.1X authentication
- **802.3ah**—IEEE 802.3ah Operation, Administration, and Maintenance (OAM) link fault management (LFM)

NOTE: If you enable L2PT for untagged OAM LFM packets, do not configure LFM on the corresponding access interface.

- **cdp**—Cisco Discovery Protocol
- **e-lmi**—Ethernet local management interface
- **gvrp**—GARP VLAN Registration Protocol
- **lacp**—Link Aggregation Control Protocol

NOTE: If you enable L2PT for untagged LACP packets, do not configure LACP on the corresponding access interface.

- **lldp**—Link Layer Discovery Protocol
- **mmrp**—Multiple MAC Registration Protocol
- **mvrp**—Multiple VLAN Registration Protocol

- **stp**—Spanning Tree Protocol, Rapid Spanning Tree Protocol, and Multiple Spanning Tree Protocol
- **udld**—Unidirectional Link Detection (UDLD)
- **vstp**—VLAN Spanning Tree Protocol
- **vtp**—VLAN Trunking Protocol

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- show ethernet-switching layer2-protocol-tunneling interface on page 1222
- show ethernet-switching layer2-protocol-tunneling statistics on page 1224
- show ethernet-switching layer2-protocol-tunneling vlan on page 1227
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395

leave-timer (MVRP)

Syntax

leave-timer *milliseconds*;

EX Series, QFX Series, QFabric

[edit protocols mvrp interface (all | *interface-name*)]

M Series, SRX Series, MX Series, T Series

[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type),
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit logical-systems *logical-system-name* protocols mvrp interface (all | *interface-name*)],

[edit protocols mvrp interface (all | *interface-name*)],

[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type)

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

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—Interval 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 (range from 300 milliseconds through 1000 milliseconds). 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.

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 MX Series Routers*
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535

- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Verifying That MVRP Is Working Correctly on page 551
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- **join-timer (MVRP) on page 1010**
- **leaveall-timer on page 1022**
- **point-to-point (MVRP) on page 1065**
- **registration on page 1095**

leaveall-timer (MVRP)

Syntax

leaveall-timer *interval*;

EX Series and QFX Series

- For platforms with ELS:

  [edit protocols mvrp],
  [edit protocols mvrp interface *interface-name*]

- For platforms without ELS:

  [edit protocols mvrp interface (all | *interface-name*)]

SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320

[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type)

EX Series, M Series, SRX Series, T Series, MX Series

[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type),
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit logical-systems *logical-system-name* protocols mvrp interface (all | *interface-name*)],
[edit protocols mvrp interface (all | *interface-name*)],
[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type)

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Hierarchy level **[edit protocols mvrp]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 3 for information about ELS.)
Statement introduced in Junos OS Release 13.1 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

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

EX Series and QFX Series:

interval—Number of seconds or milliseconds between the sending of Leave All messages.

Default: 10 seconds, or 10,000 milliseconds

SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320:

seconds—Interval between the sending of Leave All messages (range from 10 seconds through 60 seconds. 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: 60 seconds

EX Series, M Series, SRX Series, T Series, MX Series:

milliseconds—Interval between the sending of Leave All messages. 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: 10000 milliseconds

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support on page 521
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Verifying That MVRP Is Working Correctly on page 551
- **join-timer (MVRP) on page 1010**
- **leave-timer (MVRP) on page 1020**
- **point-to-point (MVRP) on page 1065**
- **registration on page 1095**

loss-priority (CoS Loss Priority)

Syntax loss-priority *level* code-points [*values*];

Hierarchy Level [edit class-of-service loss-priority-maps frame-relay-de *map-name*]

Release Information Statement introduced in Junos OS Release 9.2.

Description Map CoS values to a packet loss priority (PLP). In Junos OS, classifiers associate incoming packets with a forwarding class (FC) and PLP. PLPs allow you to set the priority for dropping packets. Typically, you mark packets exceeding some service level with a high loss priority—that is, a greater likelihood of being dropped.

Options ***level*** can be one of the following:

- **high**—Packet has high loss priority.
- **medium-high**—Packet has medium-high loss priority.
- **medium-low**—Packet has medium-low loss priority.
- **low**—Packet has low loss priority.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding Interfaces*
- Understanding Packet Loss Priorities on page 723

unicast-in-lpm

Syntax unicast-in-lpm;

Hierarchy Level [edit chassis forwarding-options lpm-profile]

Release Information Statement introduced in Junos OS Release 14.1x53-D30 for QFX Series switches.

Description For the Unified Forwarding Table feature, specify to store all unicast IPv4 and IPv6 entries with prefixes with lengths equal to or less than 64 in the table for longest prefix match (LPM) entries, thereby freeing up space in the Layer 3 host table. Only unicast entries can be moved to the LPM table. Multicast entries must be stored in the Layer 3 host table.

You can also configure this statement in conjunction with the prefix-65-127-disable statement, which allocates no memory for IPv6 prefixes with lengths in the range /65 through /127. Together, these two statements allocate more space for unicast IPv4 and IPv6 entries with prefix lengths equal to or less than 64.

NOTE: This statement is supported only on the lpm-profile.

This statement is not supported on QFX5200 switches.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding the Unified Forwarding Table*

mac (Static MAC-Based VLANs)

Syntax

```
mac mac-address {
    next-hop interface-name;
}
```

Hierarchy Level [edit ethernet-switching-options static vlan *vlan-name*]

Description Specify the MAC address to add to the Ethernet switching table.

The remaining statement is explained separately. See CLI Explorer.

Options ***mac-address***—MAC address

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure) on page 55

mac-limit

List of Syntax

Syntax (QFX Series and EX4600) on page 1027
Syntax (SRX Series and EX Series) on page 1027

Syntax (QFX Series and EX4600)

mac-limit *number*;

Syntax (SRX Series and EX Series)

mac-limit *limit* action *action*;

Hierarchy Level

[edit vlans *vlan-name*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
The short description of **interface-mac-limit** at the CLI command hierarchy is changed from **Maximum number of MAC addresses per interface (1..16383)** to **Maximum number of MAC addresses per interface (1..5120)** at the **[edit vlans vlan-name switch-options]** hierarchy level from Junos OS Release 18.2R1.

Description

Specify the maximum number of MAC addresses to be associated with a VLAN—the default is **unlimited**, which can leave the network vulnerable to flooding. Change **unlimited** to any number from 2 to the switch's maximum VLAN MAC limit. The maximum number of MAC addresses allowed in a switching table per VLAN varies depending on the EX Series switch. To see the maximum number of MAC addresses per VLAN allowed on your switch, issue the **set vlans *vlan-name* mac-limit ?** configuration-mode command.

NOTE: Do not set the **mac-limit** value to 1. The first learned MAC address is often inserted into the forwarding database automatically—for instance, for a routed VLAN interface (RVI), the first MAC address inserted into the forwarding database is the MAC address of the RVI. For aggregated Ethernet bundles (LAGs) using LACP, the first MAC address inserted into the forwarding database in the Ethernet switching table is the source address of the protocol packet. In these cases, the switch does not learn MAC addresses other than the automatic address when **mac-limit** is set to 1, and this causes problems with MAC learning and forwarding.

When the MAC limit set by this statement is reached, no more MAC addresses are added to the Ethernet switching table. You can also, optionally, have a system log entry generated when the limit is exceeded by adding the option **action log**.

NOTE: When you reconfigure the number of MAC addresses, the Ethernet switching table is not automatically cleared. Therefore, if you reduce the number of addresses from the default (unlimited) or a previously set limit, you could already have more entries in the table than the new limit allows.

Previous entries remain in the table after you reduce the number of addresses, so you should clear the Ethernet switching table for a specified interface, MAC address, or VLAN when you reduce the MAC limit. Use the command clear ethernet-switching table to clear existing MAC addresses from the table before using the mac-limit configuration statement.

Default The MAC limit is disabled, so entries are unlimited.

Options QFX Series and EX4600:

number—Maximum number of MAC addresses.
Range: 1 through 32768

NOTE: This statement is not supported on QFabric systems.

EX Series:

limit—Maximum number of MAC addresses.
Range: 1 through *switch maximum*

SRX Series:

number—Maximum number of MAC addresses.
Range: 1 through 5120

action—**Log** is the only action available. Configure **action log** to add a message to the system log when the mac-limit value is exceeded. A typical logged message looks like this:

**May 5 06:18:31 bmp-199p1-dev edwd[5665]:
ESWD_VLAN_MAC_LIMIT_EXCEEDED: vlan default mac
00:1f:12:37:af:5b (tag 40). vlan limit exceeded**

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 1510
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Configuring MAC Table Aging on Switches on page 81
- Understanding Bridging and VLANs on Switches on page 84

mac-lookup-length

Syntax mac-lookup-length *number-of-entries*;

Hierarchy Level [edit ethernet-switching-options]

Release Information Statement introduced in Junos OS Release 11.4 for EX Series switches.

Description Increase the maximum number of searchable hash indexes to mitigate situations in which hash index collisions are causing problems with the learning of MAC addresses in the forwarding database (FDB).

The FDB on EX3200, EX3300, EX4200, EX4500, EX4550, and EX6210 switches is a hash table with 8192 hash indexes (rows) of MAC addresses and four entries per hash index. When the FDB is searched, a configured hash function calculates the hash index at which to start the search. By default, after the search starts at the determined hash index, the maximum number of hash indexes that can be searched is one hash index, or four entries

NOTE: Increasing the number of hash indexes increases the chances of finding an open entry in which to add a newly learned MAC address. However, searching more hash indexes requires more bandwidth and may impact the FDB performance and line-rate traffic.

Default 4

Options ***number-of-entries***—Maximum number of searchable hash indexes in the FDB.

Range: 4, 8, 12

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Understanding Bridging and VLANs on Switches on page 84

mac-notification

Syntax

```
mac-notification {
  notification-interval seconds;
}
```

Hierarchy Level

[edit ethernet-switching-options]
[edit switch-options]

Release Information

Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Hierarchy level **[edit switch-options]** added in Junos OS Release 14.1X53-D10 for EX Series and 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. See CLI Explorer.

Default

MAC notification is disabled by default.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Non-ELS MAC Notification on page 74
- Configuring MAC Notification on Switches with ELS Support (CLI Procedure) on page 75

mac-rewrite

Syntax

```
mac-rewrite {
  interface interface-name {
    enable-all-ifl;
    protocol protocol-name;
    }
  }
}
```

Hierarchy Level [edit protocols layer2-control]

Release Information

Statement introduced in Junos OS Release 9.1.
enable-all-if statement added in Junos OS Release 13.3.
Support for PVSTP protocol introduced in Junos OS Release 13.3 for MX Series routers and EX9200 switches.
Statement introduced in Junos OS Release 14.1X53-D10 for EX4300 switches.
Statement introduced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.

Description

Enable rewriting of the MAC address for Layer 2 protocol tunneling. When a control packet for a supported protocol is received on a service provider edge port configured for Layer 2 protocol tunneling (L2PT), the multicast destination MAC address is rewritten with the predefined multicast tunneling MAC address of 01:00:0c:cd:cd:d0. The packet is transported across the provider network transparently to the other end of the tunnel, and the original multicast destination MAC address is restored when the packet is transmitted.

Refer to protocol for the list of protocols that can be configured for L2PT on different devices.

To see the protocols for which L2PT tunneling is enabled for an interface, enter the show mac-rewrite interface command.

On MX Series routers and EX9200 switches with L2PT configured, customer-facing ports should not receive packets with the L2PT MAC address as the destination address unless there is a network tolopogy or configuration error. Any such interface receiving an L2PT packet becomes “Disabled”, and must subsequently be re-enabled using the clear error mac-rewrite command.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Layer 2 Protocol Tunneling Through a Network*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389

- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398
- show mac-rewrite interface on page 1395
- clear error mac-rewrite on page 1169

mac-statistics

Syntax mac-statistics;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
[edit logical-systems *logical-system-name* switch-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* switch-options],
[edit routing-instances *routing-instance-name* protocols evpn],
[edit switch-options],
[edit switch-options],
[edit vlans *vlan-name* switch-options]

Release Information
Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options] and [edit vlans *vlan-name* switch-options] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
[edit switch-options] and [edit vlans *vlan-name* switch-options] hierarchy levels introduced in Junos OS Release 13.2 for the QFX Series.

Description (MX Series routers, EX Series switches, and QFX Series only) For bridge domains or VLANs, enable MAC accounting either for a specific bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port.

Default disabled

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*

- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port on page 200
- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*

mac-table-aging-time

Syntax mac-table-aging-time (*seconds* | unlimited);

Hierarchy Level [edit ethernet-switching-options],
[edit vlans *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 include **[edit ethernet-switching-options]** hierarchy level.

Description You configure how long MAC addresses remain in the Ethernet switching table using the **mac-table-aging-time** statement in either the **[edit ethernet-switching-options]** or the vlans hierarchy, depending on whether you want to configure it for the entire switch or only for specific VLANs.

If you specify the time as **unlimited**, entries are never removed from the table. Generally, use this setting only if the switch or the VLAN has a fairly static number of end devices; otherwise the table will eventually fill up. You can use this setting to minimize traffic loss and flooding that might occur when traffic arrives for MAC addresses that have been removed from the table.

Default Entries remain in the Ethernet switching table for 300 seconds

Options ***seconds***—Time that entries remain in the Ethernet switching table before being removed.
Range: 60 through 1,000,000 seconds
Default: 300 seconds

unlimited—Entries remain in the Ethernet switching table.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- **show ethernet-switching statistics aging on page 1237**
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Configuring MAC Table Aging on Switches on page 81
- *Controlling Authentication Session Timeouts (CLI Procedure)*
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98

mac-table-size

Syntax

```
mac-table-size limit {
  packet-action drop;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
[edit logical-systems *logical-system-name* switch-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* switch-options],
[edit switch-options],
[edit switch-options],
[edit vlans *vlan-name* switch-options]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options] and **[edit vlans *vlan-name* switch-options]** hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.
Support at the **[edit vlans *vlan-name* switch-options** hierarchy level introduced in Junos OS Release 13.2 for the QFX Series.

Description

Modify the size of the MAC address table for the bridge domain or VLAN, a set of bridge domains or VLANs associated with a trunk port, or a virtual switch. The default is 5120 MAC addresses.

NOTE: For multichassis link aggregation (MC-LAG) peers in active-active mode, configuring the **mac-table-size** statement or changing the **mac-table-size** configuration when traffic is flowing can cause the MAC entries to be out of synchronization between the two MC-LAG peers, which might result in flooding. To avoid flooding, you must either halt traffic forwarding and then configure the **mac-table-size** statement or use the **commit at** configuration statement to commit the changes at the same time in both the peer nodes.

Alternatively, if flooding does occur, you can clear the bridge MAC table on both the routers by using the **clear bridge mac-table** command. Running this

command ensures that the MAC entries are re-learned and in synchronization between both the peers.

Options

limit—Specify the maximum number of addresses in the MAC address table.

Range: 16 through 1,048,575 MAC addresses

Default: 5120 MAC addressesThere is no default MAC address limit for the **mac-table-size** statement at the **[edit switch-options]** hierarchy level. The number of MAC addresses that can be learned is only limited by the platform, 65,535 MAC addresses for EX Series switches and 1,048,575 MAC addresses for other devices.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port on page 200

mapping

List of Syntax

Syntax (EX Series) on page 1037
Syntax (QFX Series) on page 1037

Syntax (EX Series)

```
mapping (native (push | swap) | policy | tag (push | swap));
```

Syntax (QFX Series)

```
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 9.6 for EX Series switches.
Option **swap** introduced in Junos OS Release 10.0 for EX Series switches.
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.

This statement is also required if you are configuring firewall filters to map traffic from an interface to a VLAN. If you are configuring firewall filters to map traffic from an interface to a VLAN, the **mapping policy** option must be configured using this command. The firewall filter also has to be configured using the **vlan** action for a match condition in the firewall filter stanza for firewall filters to map traffic from an interface for a VLAN.

Options

For EX Series:

native—Maps untagged and priority-tagged packets to an S-VLAN.

policy—Maps the interface to a firewall filter policy to an S-VLAN.

push—Retains the incoming tag and add an additional VLAN tag instead of replacing the original tag.

swap—Swaps the incoming VLAN tag with the VLAN ID tag of the S-VLAN. Use of this option is also referred to as VLAN ID translation.

tag—Retains the incoming 802.1Q tag on the interface.

For QFX Series:

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 QFX Series Switches on page 581
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Understanding Bridging and VLANs on Switches on page 84

mapping-range

Syntax mapping-range *C-VLAN-range* (push | swap) <vlan-id-start *S-VLAN-ID*>;

Hierarchy Level [edit vlans *vlan-name* 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 QFX Series Switches on page 581
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598
- vlan-range on page 1140

match (Security Policies)

Syntax

```
match {
  application {
    [application];
    any;
  }
  destination-address {
    [address];
    any;
    any-ipv4;
    any-ipv6;
  }
  source-address {
    [address];
    any;
    any-ipv4;
    any-ipv6;
  }
  source-identity {
    [role-name];
    any;
    authenticated-user;
    unauthenticated-user;
    unknown-user;
  }
}
```

Hierarchy Level [edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name*]

Release Information Statement introduced in Junos OS Release 8.5. Statement updated with the **source-identity** option in Junos OS Release 12.1.

Description Configure security policy match criteria.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Security Policies Overview*

members

Syntax members [(all | *names* | *vlan-ids*)];

Hierarchy Level [edit interfaces (QFX Series) *interface-name* unit 0 family ethernet-switching vlan]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement updated with enhanced **?** (CLI completion feature) functionality in Junos OS Release 9.5 for EX Series switches.
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.

NOTE: The number of VLANs supported per switch varies for each model. Use the configuration-mode command **set vlans id vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because each VLAN is assigned an ID number when it is created. You can, however, exceed the recommended VLAN member maximum.

On an EX Series switch that runs Junos OS that does not support the Enhanced Layer 2 Software (ELS) configuration style, the maximum number of VLAN members allowed on the switch is 8 times the maximum number of VLANs the switch supports (vmember limit = vlan max * 8). If the switch configuration exceeds the recommended VLAN member maximum, you see a warning message when you commit the configuration. If you ignore the warning and commit such a configuration, the configuration succeeds but you run the risk of crashing the Ethernet switching process (**eswd**) due to memory allocation failure.

On an EX Series 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 the switch supports (vmember limit = vlan max * 24). If the configuration of one of these switches exceeds the recommended VLAN member maximum, a warning message appears in the system log (syslog).

Options **all**—Specifies that this trunk interface is a member of all 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: Since VLAN members are limited, specifying all could cause the number of VLAN members to exceed the limit at some point.

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—Name of one or more VLANs. VLAN IDs are applied automatically in this case.

vlan-ids—Numeric identifier of one or more VLANs. For a series of tagged VLANs, specify a range; for example, **10–20** or **10–20 23 27–30**.

NOTE: Each configured VLAN must have a specified VLAN ID to successfully commit the configuration; otherwise, the configuration commit fails.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure) for EX Series Switches with ELS support*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98
- Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure) on page 102
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Understanding Bridging and VLANs on Switches on page 84
- **show ethernet-switching interfaces on page 1215**
- **show vlans on page 1510**

mvrp

List of Syntax

Syntax (EX Series with ELS Support) on page 1043
Syntax (EX Series) on page 1043
Syntax (MX Series, EX Series, SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320) on page 1043

Syntax (EX Series with ELS Support)

```
mvrp {
  interface interface-name {
    join-timer milliseconds;
    leave-timer milliseconds;
    leaveall-timer seconds;
    registration (forbidden | normal);
  }
  join-timer milliseconds;
  leave-timer milliseconds;
  leaveall-timer seconds;
  no-attribute-length-in-pdu
  no-dynamic-vlan;
  traceoptions {
    file filename <files number > <size size> < world-readable | no-world-readable>;
    flag <flag> <disable>;
  }
}
```

Syntax (EX Series)

```
mvrp {
  add-attribute-length-in-pdu;
  disable (MVRP);
  interface (MVRP) (all | interface-name) {
    disable (MVRP);
    join-timer (MVRP) milliseconds;
    leave-timer (MVRP) milliseconds;
    leaveall-timer milliseconds;
    registration (forbidden | normal);
  }
  no-dynamic-vlan;
  traceoptions {
    file filename <files number > <size size> <no-stamp | world-readable |
      no-world-readable>;
    flag flag;
  }
}
```

Syntax (MX Series, EX Series, SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320)

```
mvrp {
  bpdu-destination-mac-address provider-bridge-group;
  join-timer (MVRP) milliseconds;
  leave-timer milliseconds;
  leaveall-timer milliseconds;
  interface (all | interface-name) {
    join-timer (MVRP) milliseconds;
    leave-timer milliseconds;
    leaveall-timer milliseconds;
    point-to-point;
```

```
    registration (forbidden | normal | restricted);
  }
  no-attribute-length-in-pdu
  no-dynamic-vlan;
  traceoptions {
    file filename <files number > <size size> <no-stamp | world-readable |
      no-world-readable>;
    flag flag;
  }
}
```

EX Series with ELS Support

[edit protocols]

EX Series and MX Series

[edit logical-systems *logical-system-name* protocols],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols] (for virtual switch instance type),

[edit protocols],

[edit routing-instances *routing-instance-name* protocols] (for virtual switch instance type),

SRX 1500, SRX 300, SRX 550M, SRX 345, SRX 340, SRX 320

[edit protocols],
[edit routing-instances *routing-instance-name* protocols] (for virtual switch instance type),

Release Information

Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches with ELS support.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

For Layer 2 networks, configure Multiple VLAN Registration Protocol (MVRP) to dynamically share VLAN information and dynamically configure needed VLANs. Maintaining VLAN configurations based on active VLANs reduces the amount of traffic traveling in the network, saving network resources. MVRP is configured on trunk interfaces.

Configure Multiple VLAN Registration Protocol (MVRP) on a trunk interface to ensure 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.

NOTE: At Junos OS Release 11.3, MVRP was updated to conform to the IEEE standard 802.1ak. This update might result in compatibility issues in mixed release networks. For details, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.

The remaining statements are explained separately. See CLI Explorer.

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

- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches with ELS Support on page 521
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Verifying That MVRP Is Working Correctly on page 551
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501

native-vlan-id

Syntax native-vlan-id *vlan-id*;

Hierarchy Level (QFX Series and EX4600)

For platforms without ELS:

[edit interfaces (QFX Series) *interface-name* unit 0 family ethernet-switching]

For platforms with ELS:

[edit interfaces (QFX Series) *interface-name*]

Hierarchy Level (ACX Series, EX Series, SRX Series, M Series, MX Series, and T Series)

[edit interfaces ge-*fpc/pic/port*],
[edit interfaces *interface-name*]

Hierarchy Level (SRX Series)

[edit interfaces*interface-name*]

Release Information

Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.5 for SRX Series.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

Configure the VLAN identifier to associate with untagged packets received on the physical interface of a trunk mode interface for the following:

- QFX Series and EX4600
- M Series routers with Gigabit Ethernet IQ PICs with SFP and Gigabit Ethernet IQ2 PICs with SFP configured for 802.1Q flexible VLAN tagging
- MX Series routers with Gigabit Ethernet DPCs and MICs, Tri-Rate Ethernet DPCs and MICs, and 10-Gigabit Ethernet DPCs and MICs and MPCs configured for 802.1Q flexible VLAN tagging
- T4000 routers with 100-Gigabit Ethernet Type 5 PIC with CFP
- EX Series switches with Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces

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, otherwise the untagged packets are dropped. 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 included with the flexible-vlan-tagging statement, untagged packets are accepted on the same mixed VLAN-tagged port and on the interfaces that are configured for Q-in-Q tunneling.

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.

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**.

NOTE: Starting in Junos OS Release 17.1R1, you can send untagged traffic without a native VLAN ID to the remote end of the network. To do this, remove the native VLAN ID from the untagged traffic configuration by setting the **no-native-vlan-insert** statement. If you do not configure this statement, the native VLAN ID is added to the untagged traffic.

Default By default, the untagged packets are dropped. That is, if you do not configure the **native-vlan-id** option, the untagged packets are dropped.

Options ***vlan-id***—Numeric identifier of the VLAN.

Range: 1 through 4094

number—VLAN ID number.

Range: (ACX Series routers, SRX Series devices and EX Series switches) 0 through 4094.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- Understanding Bridging and VLANs on Switches on page 84
- *Enabling VLAN Tagging*
- *Configuring Access Mode on a Logical Interface*
- Configuring the Native VLAN Identifier on Switches With ELS Support (CLI Procedure) on page 196
- *Understanding Interfaces*
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- *no-native-vlan-insert*
- *Sending Untagged Traffic Without VLAN ID to Remote End*
- **show ethernet-switching interfaces on page 1215**
- **show vlans on page 1510**
- **flexible-vlan-tagging on page 960**
- *Junos OS Network Interfaces Configuration Guide*

next-hop (Static MAC-Based VLANs)

Syntax next-hop *interface-name;*

Hierarchy Level [edit ethernet-switching-options static vlan *vlan-name* mac *mac-address*]

Release Information Statement introduced in Junos OS Release 11.1 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Specify the next hop for the indicated Ethernet node.

Options ***interface-name***—Name of the next-hop interface.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure) on page 55

no-attribute-length-in-pdu

Syntax no-attribute-length-in-pdu;

Hierarchy Level [edit protocols mvrp]

Release Information Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description Include an extra byte in protocol data units (PDUs) sent by the Multiple VLAN Registration Protocol (MVRP). You can disable the extra byte to address a compatibility issue between MVRP in Junos OS Releases 13.2 and later for EX Series switches with support for the Enhanced Layer 2 Software (ELS), which includes the extra byte, and MVRP in Junos OS Releases 11.3 and later for EX Series switches that do not support ELS, which does not include the extra byte. If this compatibility issue arises, the ELS version of MVRP does not recognize PDUs without the extra byte sent by the non-ELS version of MVRP.

You can recognize an MVRP version compatibility issue by observing the switch running the ELS version of MVRP. Because a switch running the ELS version of MVRP cannot interpret an unmodified PDU from a switch running the non-ELS version of MVRP, the switch will not add VLANs from the non-ELS version of MVRP. When you execute the command **show mvrp statistics** in the ELS version of MVRP, the values for **Received Join Empty** and **Received Join In** will incorrectly display zero, even though the value for the **Received MVRP PDUs without error** has been increased. Another indication that MVRP is having a version compatibility issue is that unexpected VLAN activity, such as multiple VLAN creation, takes place on the switch running the ELS version of MVRP.

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) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501

no-dynamic-vlan

Syntax no-dynamic-vlan;

Hierarchy Level [edit protocols mvrp]

[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type)

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description Disable the dynamic creation of VLANs using Multiple VLAN Registration Protocol (MVRP) for interfaces participating in MVRP.

Dynamic VLAN configuration can be enabled on an interface independent of MVRP. The MVRP dynamic VLAN configuration setting does not override the interface configuration dynamic VLAN configuration setting. If dynamic VLAN creation is disabled on the interface in the interface configuration, no dynamic VLANs are created on the interface, including dynamic VLANs created using MVRP.

This option can be applied globally; it cannot be applied per interface.

Default If MVRP is enabled, the dynamic creation of VLANs as a result of MVRP protocol exchange messages is enabled.

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) on Switches on page 504
- Configuring Multiple VLAN Registration Protocol (MVRP) on Switches on page 504
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501

no-gratuitous-arp-request

Syntax no-gratuitous-arp-request;

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Configure the switch not to respond to gratuitous ARP requests. You can disable responses to gratuitous ARP requests on Layer 2 Ethernet switching interfaces, and integrated routing and bridging (IRB) interfaces or routed VLAN interfaces (RVIs). (On EX Series switches that use Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style, the feature is known as an IRB interface. On EX Series switches that use Junos OS that does not support ELS, the feature is known as an RVI.)

Default Gratuitous ARP responses are enabled on all Ethernet switching interfaces, and IRB interfaces or RVIs.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Proxy ARP on an EX Series Switch on page 629
- Configuring Proxy ARP on Switches (CLI Procedure) on page 627
- Configuring Proxy ARP on Devices with ELS Support (CLI Procedure) on page 628

no-local-switching

Syntax no-local-switching

Hierarchy Level [edit vlans *vlan-name*]

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 Specify that access ports in this VLAN domain do not forward packets to each other. You use this statement with primary VLANs and isolated secondary VLANs.

Required Privilege Level system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
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 QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275

no-mac-learning

Syntax

no-mac-learning;

QFX Series and EX4600

For QFX Series and EX4600 platforms without ELS:

[edit ethernet-switching-options interfaces *interface-name*]

For QFX Series and EX4600 platforms with ELS:

[edit vlans *vlan-name* switch-options]

QFX Series per VLAN

[edit vlans *vlan-name*]

[edit vlans *vlan-name* switch-options]

EX Series Q-in-Q Interfaces

[edit ethernet-switching-options interfaces *interface-name*]

EX Series and SRX Series Q-inQ Vlans

[edit vlans *vlan-name*]

ACX Series, MX Series, EX Series with ELS support, M Series, T Series

[edit bridge-domains *bridge-domain-name* bridge-options],
[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options],
[edit logical-systems *logical-system-name* switch-options],
[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols evpn],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name*],
[edit routing-instances *routing-instance-name* switch-options],
[edit switch-options],
[edit switch-options],
[edit switch-options interface *interface-name*],
[set vlans *vlan-name* switch-options]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy

supported this statement only for a VPLS instance or bridge domain configured within a virtual switch.
Statement introduced in Junos OS Release 9.5 for EX Series switches.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
[edit switch-options], **[edit switch-options interface *interface-name*]**, **[edit vlans *vlan-name* switch-options]**, and **[edit vlans *vlan-name* switch-options interface *interface-name*]** hierarchy levels introduced in Junos OS Release 12.3 R2 for EX Series switches.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
Hierarchy levels **[edit switch-options interface *interface-name*]** and **[edit vlans *vlan-name* switch-options]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches.

Description For QFX Series, EX Series switches and SRX Series devices, disables MAC address learning for the specified VLAN.

For QFX Series and EX4600, 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.

For EX Series switches' Q-in-Q interfaces, disables 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.

For MX Series routers and EX Series switches with ELS support, disables MAC learning for a virtual switch, for a bridge domain or VLAN, for a specific logical interface in a bridge domain or VLAN, or for a set of bridge domains or VLANs associated with a Layer 2 trunk port. On platforms that support EVPNs, you can disable MAC learning on an EVPN.

NOTE: When MAC learning is disabled for a VPLS routing instance, traffic is not load-balanced and only one of the equal-cost next hops is used.

Default MAC learning is enabled.

Required Privilege Level
system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Understanding Bridging and VLANs on Switches on page 84
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Understanding Q-in-Q Tunneling and VLAN Translation on page 554
- Configuring Q-in-Q Tunneling on EX Series Switches (CLI Procedure) on page 582

node-id

Syntax node-id *mac-address*;

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information Statement introduced in Junos OS Release 9.4.

Description For EX Series switches and QFX Series switches, node-id is not configurable.

For MX Series routers, optionally specify the MAC address of a node in the protection group. If this statement is not included, the router assigns the node's MAC address.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435

notification-interval

Syntax notification-interval *seconds*;

Hierarchy Level [edit ethernet-switching-options mac-notification]
[edit switch-options mac-notification]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Hierarchy level **[edit switch-options]** added in Junos OS Release 14.1X53-D10 for EX Series and 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 will be 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 system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Non-ELS MAC Notification on page 74
- Configuring Non-ELS MAC Notification on page 74
- Configuring MAC Notification on Switches with ELS Support (CLI Procedure) on page 75

num-65-127-prefix

Syntax num-65-127-prefix *number*;

Hierarchy Level [edit chassis (QFX Series) forwarding-options *profile-name*]

Release Information Statement introduced in Junos OS Release 13.2 for QFX Series switches.
Support for QFX5200 Series switches introduced in Junos OS Release 15.1X53-D30.

Description For the Unified Forwarding Table (UFT) feature, specify how much forwarding table memory to allocate for IPv6 entries with prefix lengths in the range of /65 through /127. The ability to allocate flexibly the memory for IPv6 entries with prefixes in this range extends the use of this memory space to accommodate the appropriate mix of longest-prefix match (LPM) entries that best suits your network. The LPM table stores IPv4 unicast prefixes, IPv6 prefixes with lengths equal to or less than 64, and IPv6 prefixes with lengths from 65 through 127. With this option, you can increase, decrease, or allocate no memory for IPv6 prefixes with lengths from 65 through 127, depending on which version of Junos OS you are using.

NOTE: This statement is supported only for the following forwarding table memory profiles: l2-profile-one, l2-profile-three, l2-profile-two, and l3-profile. Do not use this statement with the custom-profile or the lpm-profile statements.

NOTE: The values you can configure are different depending on the version of Junos OS you are using.

Options **number**—Specify a numerical value.

Range: (Junos OS Release 13.2x51-D10 only) 1 through 128. Each increment represents 16 IPv6 prefixes with lengths in the range of /65 through /127, for a total maximum of 2,058 prefixes (16 x 128 = 2,048).

Default: 1 (16 IPv6 prefixes with lengths in the range of /65 through /127).

Range: (Junos OS Release 13.2X51-D15 or later) 0 through 4. Each increment allocates memory for 1,000 IPv6 prefixes with lengths in the range of /65 through /127, for a maximum of 4,000 such IPv6 prefixes.

Default: 1 (1,000 IPv6 prefixes with lengths in the range of /65 through /127).

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Unified Forwarding Table on Switches on page 37

output-vlan-map

Syntax

```
output-vlan-map {
  (pop | pop-pop | pop-swap | push | push-push | swap | swap-push | swap-swap);
  inner-tag-protocol-id tpid;
  inner-vlan-id number;
  tag-protocol-id tpid;
  vlan-id number;
}
```

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.
pop-pop, **pop-swap**, **push-push**, **swap-push**, and **swap-swap** statements added in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

For EX Series switches, defines the rewrite operation to be applied to outgoing frames.

For MX Series routers and NFX Series devices' Gigabit Ethernet IQ and 10-Port 10-Gigabit Ethernet SFPP interfaces only, defines the rewrite operation to be applied to outgoing frames on this logical interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Stacking and Rewriting Gigabit Ethernet VLAN Tags*
- input-vlan-map on page 984

packet-action

Syntax packet-action *action*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
[edit bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface *interface-name* interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],
[edit logical-systems *logical-system-name* switch-options interface-mac-limit *limit*],
[edit protocols l2-learning global-mac-limit *limit*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name* interface-mac-limit *limit*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface-mac-limit *limit*],
[edit routing-instances *routing-instance-name* protocols evpn interface-mac-limit (VPLS)],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name* interface-mac-limit (VPLS)],
[edit routing-instances *routing-instance-name* protocols evpn mac-table-size *limit*],
[edit routing-instances *routing-instance-name* switch-options interface *interface-name* interface-mac-limit *limit*],
[edit routing-instances *routing-instance-name* switch-options interface-mac-limit *limit*],
[edit switch-options interface-mac-limit *limit*],
[edit switch-options interface *interface-name* interface-mac-limit *limit*],
[edit switch-options interface-mac-limit *limit*],
[edit switch-options interface *interface-name* interface-mac-limit *limit*],
[edit switch-options interface-mac-limit *limit*],
[edit switch-options mac-table-size *limit*],
[edit switch-options interface *interface-name* interface-mac-limit *limit*],
[edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*],
[edit vlans *vlan-name* switch-options interface-mac-limit *limit*],
[edit vlans *vlan-name* switch-options mac-table-size *limit*]
[edit vlans *vlan-name* switch-options interface-mac-limit *limit*],
[edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*],
[edit vlans *vlan-name* switch-options mac-table-size *limit*]

Release Information Statement introduced in Junos OS Release 8.4.
Support for the **switch-options** statement added in Junos OS Release 9.2.
Support for top-level configuration for the **virtual-switch** type of routing instance added in Junos OS Release 9.2. In Junos OS Release 9.1 and earlier, the routing instances hierarchy

supported this statement only for a VPLS instance or a bridge domain configured within a virtual switch.
Support for logical systems added in Junos OS Release 9.6.
[edit switch-options interface *interface-name* interface-mac-limit *limit*], [edit switch-options interface-mac-limit *limit*], [edit switch-options mac-table-size *limit*], [edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit *limit*], [edit vlans *vlan-name* switch-options interface-mac-limit *limit*], and [edit vlans *vlan-name* switch-options mac-table-size *limit*] hierarchy levels introduced in Junos OS Release 12.3R2 for EX Series switches.
Support for EVPNs introduced in Junos OS Release 13.2 on MX Series 5G Universal Routing Platforms.
Support at the **[edit switch-options interface *interface-name* interface-mac-limit *limit*]** hierarchy level and hierarchy levels under **[edit vlans *vlan-name*]** introduced in Junos OS Release 13.2X50-D10 for EX Series switches and Junos OS Release 13.2 for the QFX Series.

Description Specify the action taken when packets with new source MAC addresses are received after the MAC address limit is reached. If this statement is not configured, packets with new source MAC addresses are forwarded by default.

NOTE: The packet-action statement is not supported on the QFX10002-60C switch.

Default

NOTE: On a QFX Series Virtual Chassis, if you include the **shutdown** option at the **[edit vlans *vlan-name* switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level and issue the **commit** operation, the system generates a commit error. The system does not generate an error if you include the **shutdown** option at the **[edit switch-options interface *interface-name* interface-mac-limit packet-action]** hierarchy level.

Disabled. The default is for packets for new source MAC addresses to be forwarded after the MAC address limit is reached.

Options

drop—Drop packets with new source MAC addresses, and do not learn the new source MAC addresses.

NOTE: On QFX10000 switches, if you include the drop option, you cannot configure unicast reverse-path forwarding (URFP) on integrated routing and bridging (IRB) and MAC limiting on the same interface. If you have an MC-LAG configuration, you cannot configure MAC limiting on the interchassis link (ICL) interface.

drop-and-log—(EX Series switches and QFX Series only) Drop packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

log—(EX Series switches and QFX Series only) Hold packets with new source MAC addresses, and generate an alarm, an SNMP trap, or a system log entry.

none—(EX Series switches and QFX Series only) Forward packets with new source MAC addresses, and learn the new source MAC address.

shutdown—(EX Series switches and QFX Series only) Disable the specified interface, and generate an alarm, an SNMP trap, or a system log entry.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring EVPN Routing Instances*
- *Configuring EVPN Routing Instances on EX9200 Switches*
- Configuring MAC Limiting (CLI Procedure) on page 58
- *Configuring Persistent MAC Learning (CLI Procedure)*
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port on page 200

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 on QFX Switches Using MVRP on page 516

peer-selection-service

Syntax

```
peer-selection-service {
  command binary-file-path;
  disable;
  failover (alternate-media | other-routing-engine);
}
```

Hierarchy Level [edit system processes]

Release Information Statement introduced in Junos OS Release 8.5.

Description Enable the peer selection service process.

Options

- **command *binary-file-path***—Path to the binary process.
- **disable**—Disable the peer selection service process.
- **failover**—Configure the device to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.
  - **alternate-media**—Configure the device to switch to backup media that contains a version of the system if a software process fails repeatedly.
  - **other-routing-engine**—Instruct the secondary Routing Engine to take mastership if a software process fails. If this statement is configured for a process, and that process fails four times within 30 seconds, then the device reboots from the secondary Routing Engine.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Interfaces Feature Guide for Security Devices

pgcp-service

Syntax

```
pgcp-service {
  command binary-file-path;
  disable;
  failover (alternate-media | other-routing-engine);
}
```

Hierarchy Level [edit system processes]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the Packet Gateway Control Protocol (PGCP) that is required for the border gateway function (BGF) feature.

Options

- **command *binary-file-path***—Path to the binary process.
- **disable**—Disable the Packet Gateway Control Protocol (PGCP) process.
- **failover**—Configure the device to reboot if the software process fails four times within 30 seconds, and specify the software to use during the reboot.
  - **alternate-media**—Configure the device to switch to backup media that contains a version of the system if a software process fails repeatedly.
  - **other-routing-engine**—Instruct the secondary Routing Engine to take mastership if a software process fails. If this statement is configured for a process, and that process fails four times within 30 seconds, the device reboots from the secondary Routing Engine.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *restart (Reset)*

point-to-point (MVRP)

Syntax point-to-point;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mvrp interface (all | *interface-name*)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type),

[edit protocols mvrp interface (all | *interface-name*)],

[edit routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type)

Release Information Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description (Optional) For Multiple VLAN Registration Protocol (MVRP) configurations, configure an interface to be recognized as a point-to-point connection. If specified, a point-to-point subset of the MRP state machine is used to provide a simpler and more efficient method to accelerate convergence on the network. Point-to-point must be enabled after enabling MVRP for the interface to be recognized as a point-to-point connection.

Default MVRP is disabled by default.

point-to-point 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 MX Series Routers*
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Verifying That MVRP Is Working Correctly on page 551
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Understanding Multiple VLAN Registration Protocol (MVRP) on page 496
- join-timer on page 1010
- leaveall-timer on page 1022
- leave-timer on page 1020
- registration on page 1095

policy (Security Policies)

Syntax

```
policy policy-name {
  description description;
  match {
    application {
      [application];
      any;
      junos-twamp;
    }
    destination-address {
      [address];
      any;
      any-ipv4;
      any-ipv6;
    }
    source-address {
      [address];
      any;
      any-ipv4;
      any-ipv6;
    }
    source-identity {
      [role-name];
      any;
      authenticated-user;
      unauthenticated-user;
      unknown-user;
    }
  }
  scheduler-name scheduler-name;
  then {
    count {
      alarm {
        per-minute-threshold number;
        per-second-threshold number;
      }
    }
    deny;
    log {
      session-close;
      session-init;
    }
    permit {
      application-services {
        application-firewall {
          rule-set rule-set-name;
        }
        application-traffic-control {
          rule-set rule-set-name;
        }
        gprs-gtp-profile profile-name;
        gprs-sctp-profile profile-name;
        idp;
```

```
        redirect-wx | reverse-redirect-wx;
        ssl-proxy {
          profile-name profile-name;
        }
        uac-policy {
          captive-portal captive-portal;
        }
        utm-policy policy-name;
      }
      destination-address {
        drop-translated;
        drop-untranslated;
      }
      firewall-authentication {
        pass-through {
          access-profile profile-name;
          client-match user-or-group-name;
          web-redirect;
        }
        user-firewall {
          access-profile profile-name;
          domain domain-name
          ssl-termination-profile profile-name;
        }
        web-authentication {
          client-match user-or-group-name;
        }
      }
      services-offload;
      tcp-options {
        initial-tcp-mss mss-value;
        reverse-tcp-mss mss-value;
        sequence-check-required;
        syn-check-required;
      }
      tunnel {
        ipsec-group-vpn group-vpn;
        ipsec-vpn vpn-name;
        pair-policy pair-policy;
      }
    }
    reject;
  }
}
```

Hierarchy Level [edit security policies from-zone *zone-name* to-zone *zone-name*]

Release Information Statement introduced in Junos OS Release 8.5. The **services-offload** option added in Junos OS Release 11.4. Statement updated with the **source-identity** option and the **description** option added in Junos OS Release 12.1. Support for the **user-firewall** option added in Junos OS Release 12.1X45-D10. Support for the **initial-tcp-mss** and **reverse-tcp-mss** options added in Junos OS Release 12.3X48-D20.
The **junos-twamp** application is introduced in Junos OS Release 18.2R1.

Description Define a security policy.

Options ***policy-name***—Name of the security policy.

—

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Configuring SSL Forward Proxy*
- *Security Policies Overview*

pop

Syntax pop;

Hierarchy Level
[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

NOTE: On EX4300 switches, **pop** is not supported at the [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map] hierarchy level.

For Gigabit Ethernet IQ, 10-Gigabit Ethernet IQ2, and IQ2-E interfaces; 10-Gigabit Ethernet LAN/WAN PIC; aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces; 100-Gigabit Ethernet Type 5 PIC with CFP; and Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces, specify the VLAN rewrite operation to remove a VLAN tag from the top of the VLAN tag stack. The outer VLAN tag of the frame is removed.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Removing a VLAN Tag*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

pop-pop

Syntax pop-pop;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description For Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP, and for 10-Gigabit Ethernet SFP interfaces on EX Series switches, specify the VLAN rewrite operation to remove both the outer and inner VLAN tags of the frame.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Removing the Outer and Inner VLAN Tags*

pop-swap

Syntax pop-swap;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Specify the VLAN rewrite operation to remove the outer VLAN tag of the frame, and replace the inner VLAN tag of the frame with a user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame.

You can use this statement on Gigabit Ethernet IQ, IQ2, IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Removing the Outer VLAN Tag and Rewriting the Inner VLAN Tag*

port-mode

Syntax port-mode (access | tagged-access | trunk);

Hierarchy Level [edit interfaces (QFX Series) *interface-name* unit *logical-unit-number* family ethernet-switching]

Release Information 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

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 "Using the Enhanced Layer 2 Software CLI" on page 3.

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.

NOTE: The number of VLANs supported per switch varies for each model. Use the configuration-mode command **set vlans id vlan-id ?** to determine the maximum number of VLANs allowed on a switch. You cannot exceed this VLAN limit because each VLAN is assigned an ID number when it is created. You can, however, exceed the recommended VLAN member maximum. To determine the maximum number of VLAN members allowed on a switch, multiply the VLAN maximum for the switch times 8 (vmember limit = vlan max * 8).

If a switch configuration exceeds the recommended VLAN member maximum, you see a warning message when you commit the configuration. If you ignore the warning and commit such a configuration, the configuration succeeds

but you run the risk of crashing the Ethernet switching process (eswd) due to memory allocation failure.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98
- *Junos OS Ethernet Interfaces Configuration Guide*

preempt-cutover-timer

Syntax preempt-cutover-timer *seconds*;

Hierarchy Level

- For platforms with ELS:

  [edit switch-options redundant-trunk-group group *name*]
  [edit interfaces *name* aggregated-ether-options lacp link-protection rtg-config]
  [edit interfaces *name* aggregated-ether-options link-protection rtg-config]

- For platforms without ELS:

  [edit ethernet-switching-optionsredundant-trunk-group group *name*]

Release Information Statement introduced in Junos OS Release 11.1 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 3 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description Change the length of time that a re-enabled primary link waits to take over from an active secondary link in a redundant trunk group (RTG).

Default If you do not change the time with the **preempt-cutover-timer** statement, a re-enabled primary link takes over from the active secondary link after 1 second.

Options ***seconds***—Number of seconds that the primary link waits to take over from the active secondary link.

Range: 1 through 600 seconds

Required Privilege Level admin—To view this statement in the configuration.
admin-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support on page 613
- *Q-in-Q Support on Redundant Trunk Links Using LAGs with Link Protection*

prefix-65-127-disable

Syntax prefix-65-127-disable;

Hierarchy Level [edit chassis (QFX Series) forwarding-options lpm-profile]

Release Information Statement introduced in Junos OS Release 13.2X51-D15 for QFX Series switches.
Support introduced in Junos OS Release 15.1X53-D30 for QFX5200 Series switches.
Support introduced in Junos OS Release 18.1R1 for QFX5200-48C and QFX5210 switches.

Description For the Unified Forwarding Table (UFT) feature, specify not to allocate any memory for IPv6 prefixes with lengths in the range /65 through /127 for longest-prefix-match (LPM) entries. Doing so increases the memory available for LPM entries for IPv4 unicast prefixes and IPv6 prefixes with lengths equal to or less than 64. The maximum default value for LPM entries is 16,000 IPv6 prefixes of all lengths.

In an environment where the switch is being used in the core of the network, for example, it might not need to store IPv6 prefixes with lengths in the range /65 through /127. IPv6 prefixes of this type are not typically used in the core.

NOTE: When using this statement, IPv6 prefixes within the range /65 through /127 will still appear in the routing table, but will *not* be installed in the forwarding table; therefore, matching traffic will be dropped. Note further that if a default route is configured, traffic will be forwarded, though it will be sent through the RE and rate-limited.

NOTE: On QFX5100 switches, when you configure this statement, the maximum number of LPM IPv6 entries with prefix lengths equal to or less than 64 increases to 128,000. On the QFX5200 switch, when you configure this statement, the maximum number of IPv6 entries with prefix lengths equal to or less 64 that are allocated in the LPM table increases to 98,0000.

NOTE: This statement is supported only with the lpm-profile. No other profile is supported.

The effects of this statement can be seen on a QFX5100 as follows:

```
[edit]
user@host# set chassis forwarding-options lpm-profile prefix-65-127-disable

[edit]
```

```
user@host# commit
configuration check succeeds
commit complete


[edit]
user@host# run show chassis forwarding-options
fpc0:
--------------------------------------------------------------------------
Current UFT Configuration:
lpm-profile. (MAC: 32K L3-host: 16K LPM: 128K)
prefix-65-127 = disable


[edit]
user@host# run show pfe route summary hw
Slot 0
================ fpc0 ================

Unit: 0
Profile active: lpm-profile
Type            Max       Used      Free      % free
-----------------------------------------------------
IPv4 Host       16384     20        16354     99.82
IPv4 LPM        131072    5         131065    99.99
IPv4 Mcast      8192      0         8177      99.82

IPv6 Host       8192      5         8177      99.82
IPv6 LPM(< 64)  131072    2         131065    99.99
IPv6 LPM(> 64) 0    0    0    0.00
IPv6 Mcast      4096      0         4089      99.83
```

Options **None**—This statement has no options.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Configuring the Unified Forwarding Table on Switches on page 37
- *Understanding the Unified Forwarding Table*

primary-vlan

Syntax primary-vlan *vlan-name*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.3 for EX Series switches.
Statement updated with enhanced **?** (CLI completion feature) functionality in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description For a private VLAN (PVLAN), configure the primary VLAN. The primary VLAN is always tagged.

- If the PVLAN is configured on a single switch, do not assign a tag to the community VLANs.
- If the PVLAN is configured to span multiple switches, you must assign tags to the community VLANs also.

For a community VLAN, configure the primary 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 system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
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 QFX Switch on page 269
- Creating a Private VLAN on a Single EX Series Switch (CLI Procedure) on page 271
- Example: Configuring a Private VLAN on a Single EX Series Switch on page 284
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275
- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*
- Example: Configuring a Private VLAN Spanning Multiple EX Series Switches on page 326
- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*

private-vlan

Syntax private-vlan (isolated | community) vlan-id *number*;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 14.1X53-D10 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.

Description Configure a secondary VLAN (either an isolated VLAN or a community VLAN) within a private VLAN (PVLAN) and specify a VLAN ID for that secondary VLAN. This statement essentially converts a VLAN into a PVLAN, by carving out discrete subdomains (secondary VLANs) within the primary VLAN. You must specify a VLAN ID for each secondary PVLAN.

NOTE: After you have configured the secondary VLAN, you must also configure its association with a specific primary VLAN. See isolated-vlan and community-vlan for additional information.

Options

- **isolated** — The VLAN specified by *vlan-name* is defined as an *isolated* VLAN and a VLAN-ID is assigned to it. An isolated VLAN receives packets only from the primary VLAN and forwards frames upstream to the primary VLAN. The VLAN name is optional. The VLAN ID is required.
- **community** — The VLAN specified by *vlan-name* is defined as community VLAN and a VLAN-ID is assigned to it. A *community* VLAN used to transport frames among members of a community, which is a subset of users within the VLAN, and to forward frames upstream to the primary VLAN. The VLAN name is optional. The VLAN ID is required.

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 with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277

profile (Access)

Syntax

```
profile profile-name {
  accounting {
    accounting-stop-on-access-deny;
    accounting-stop-on-failure;
    coa-immediate-update;
    duplication;
    immediate-update;
    order [accounting-method];
    statistics (time | volume-time);
    update-interval minutes;
  }
  accounting-order [accounting-method];
  address-assignment pool pool-name;
  authentication-order [ldap | none | password | securid];
  authorization-order [jsrc];
  client client-name {
    chap-secret chap-secret;
    client-group [ group-names ];
    firewall-user {
      password password;
    }
    no-rfc2486;
    pap-password pap-password;
    x-auth ip-address;
  }
  client-name-filter {
    count number;
    domain-name domain-name;
    separator special-character;
  }
  ldap-options {
    assemble {
      common-name common-name;
    }
    base-distinguished-name base-distinguished-name;
    revert-interval seconds;
    search {
      admin-search {
        distinguished-name distinguished-name;
        password password;
      }
      search-filter search-filter-name;
    }
  }
  ldap-server server-address {
    port port-number;
    retry attempts;
    routing-instance routing-instance-name;
    source-address source-address;
    timeout seconds;
  }
  provisioning-order (gx-plus | jsrc);
```

```
  service {
    accounting-order {
      activation-protocol;
      radius;
    }
  }
  session-options {
    client-group [group-name];
    client-idle-timeout minutes;
    client-session-timeout minutes;
  }
}
```

Hierarchy Level [edit access]

Release Information Statement introduced in Junos OS Release 10.4.

Description Create a profile containing a set of attributes that define device management access.

Required Privilege Level access—To view this statement in the configuration.
access-control—To add this statement to the configuration.

Related Documentation
- *Understanding Interfaces*
- *Understanding User Authentication for Security Devices*
- Ethernet Switching and Layer 2 Transparent Mode Overview on page 25

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 QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275

protection-group

Syntax

```
protection-group {
    ethernet-ring ring-name {
      data-channel {
        vlan number
      }
      east-interface {
        control-channel channel-name {
          vlan number;
          interface name interface-name
        }
      }
      guard-interval number;
      node-id mac-address;
      restore-interval number;
      ring-protection-link-owner;
      non-revertive;
      wait-to-block-interval number;
      major-ring-name name;
      propagate-tc;
      compatibility-version (1|2);
      ring-id number;
      non-vc-mode;
      dot1p-priority number;
      west-interface {
        control-channel channel-name {
          vlan number;
          interface name interface-name
        }
        virtual-control-channel {
          west-interface name;
          east-interface name;
        }
      }
    }
  control-vlan (vlan-id | vlan-name);
    east-interface {
      node-id mac-address;
      control-channel channel-name {
        vlan number;
        interface name interface-name
      }
      interface-none
      ring-protection-link-end;
    }
  }
    control-channel channel-name {
      vlan number;
      interface name interface-name
    }
  }
    data-channel {
      vlan number
```

```
    }
    guard-interval number;
    node-id mac-address;
    restore-interval number;
    ring-protection-link-owner;
      west-interface {
        node-id mac-address;
        control-channel channel-name {
          vlan number;
          interface name interface-name
        }
        interface-none
        ring-protection-link-end;
      }
      control-channel channel-name {
        vlan number;
        interface name interface-name
      }
    }
  }
  guard-interval number;
  restore-interval number;
  traceoptions {
    file filename <no-stamp> <world-readable | no-world-readable> <replace> <size size>;
    flag flag;
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description Configure Ethernet ring protection switching.

The statements are explained separately. All statements apply to MX Series routers. EX Series switches do not assign **node-id** and use **control-vlan** instead of **control-channel**.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- *Example: Configuring Load Balancing Within Ethernet Ring Protection for MX Series Routers*
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435

protocol

List of Syntax

Syntax (MX Series Routers)

```
protocol (cdp | pvstp | stp | vtp);
```

Syntax (EX2300, EX3400, EX4300 and EX4600 Switches)

```
protocol (cdp | elmi | gvrp | ieee8021x | ieee8023ah | lacp | lldp | mmrp | mvrp | stp | udld |
    vstp | vtp);
```

Syntax (EX2300 Multigigabit Model Switches)

```
protocol (cdp | gvrp | ieee8023ah | lacp | lldp | mvrp | stp | vstp | vtp);
```

Syntax (EX9200 Switches)

```
protocol (cdp | elmi | gvrp | ieee8021x | ieee8023ah | lacp | lldp | mmrp | mvrp | pvstp | stp |
    udld | vtp);
```

Hierarchy Level

[edit logical-systems *name* protocols layer2-control mac-rewrite interface *interface-name*], [edit protocols layer2-control mac-rewrite interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 9.1.
Statement introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Support for PVST/PVST+ introduced in Junos OS Release 13.3 for MX Series routers and EX9200 switches.
Statement introduced in Junos OS Release 14.1X53-D10 for EX4300 switches
Statement introduced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Support for E-LMI, IEEE 802.1X, MMRP, and UDLD introduced in Junos OS Release 17.3R1 for EX4300 switches.
Support for E-LMI, IEEE 802.1X, MMRP, and UDLD introduced in Junos OS Release 18.2R1 for EX2300 and EX3400 switches.
Support for E-LMI, GVRP, IEEE 802.1x, IEEE 802.3AH, LACP, LLDP, MMRP, MVRP, and UDLD introduced in Junos OS Release 17.3R1 for EX9200 switches.
Statement introduced in Junos OS Release 17.4R1 for EX4600 switches.

Description

Configure the protocol to be tunneled on an interface for Layer 2 protocol tunneling (L2PT). To enable tunneling multiple protocols, include multiple **protocol** statements.

Not all protocols listed in the Options section can be tunneled on all devices. The Syntax and Release Information sections list the available options for the protocols that can be tunneled by different devices as of a particular Junos OS release.

When a control packet for a supported protocol is received on a service provider edge port configured for Layer 2 protocol tunneling (L2PT), the multicast destination MAC address is rewritten with the predefined multicast tunneling MAC address of 01:00:0c:cd:cd:d0. The packet is transported across the provider network transparently

to the other end of the tunnel, and the original multicast destination MAC address is restored when the packet is transmitted.

Options

cdp—Tunnel the Cisco Discovery Protocol (CDP).

elmi—Tunnel Ethernet Local Management Interface (E-LMI) packets.

gvrp—Tunnel Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP) packets.

ieee8021x—Tunnel IEEE 802.1X authentication packets.

ieee8023ah—Tunnel IEEE 802.3AH Operation, Administration, and Maintenance (OAM) link fault management (LFM) packets.

lacp—Tunnel Link Aggregation Control Protocol (LACP) packets.

lldp—Tunnel Link Layer Discovery Protocol (LLDP) packets.

mmrp—Tunnel Multiple MAC Registration Protocol (MMRP) packets.

mvrp—Tunnel Multiple VLAN Registration Protocol (MVRP) packets.

pvstp—Tunnel VLAN Spanning Tree Protocol (VSTP), Per-VLAN Spanning Tree (PVST), and Per-VLAN Spanning Tree Plus (PVST+) Protocol packets.

stp—Tunnel packets for all versions of Spanning-Tree Protocols.

udld—Tunnel Unidirectional Link Detection (UDLD) packets.

vstp—Tunnel VLAN Spanning Tree Protocol (VSTP) packets.

vtp—Tunnel VLAN Trunking Protocol (VTP) packets.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Layer 2 Protocol Tunneling Through a Network*
- *Layer 2 Protocol Tunnel Configuration Guidelines*
- *Configuring Layer 2 Protocol Tunneling*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398

protocols (Fabric)

Syntax

```
protocols {
  fabric-control {
    graceful-restart {
      restart-time seconds;
      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. See CLI Explorer.

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*

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 on page 634
- Configuring Proxy ARP on Switches (CLI Procedure) on page 627
- Example: Configuring Proxy ARP on an EX Series Switch on page 629
- *Configuring Gratuitous ARP*

push

Syntax push;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description

NOTE: On EX4300 switches, **push** is not supported at the [edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map] hierarchy level.

Specify the VLAN rewrite operation to add a new VLAN tag to the top of the VLAN stack. An outer VLAN tag is pushed in front of the existing VLAN tag.

You can use this statement on Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces; 10-Gigabit Ethernet LAN/WAN PIC; aggregated Ethernet interfaces using Gigabit Ethernet IQ interfaces; 100-Gigabit Ethernet Type 5 PIC with CFP; and Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces.

If you include the **push** statement in the configuration, you must also include the pop statement at the **[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map]** hierarchy level.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Stacking a VLAN Tag*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

push-push

Syntax push-push;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Specify the VLAN rewrite operation to push two VLAN tags in front of the frame.

You can use this statement on Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, on aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *Stacking Two VLAN Tags*

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 QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275

pvlan-trunk

Syntax pvlan-trunk;

Hierarchy Level [edit vlans *vlan-name* vlan-id *number* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 10.4 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure an interface to be the trunk port, connecting switches that are configured with a private VLAN (PVLAN) across these switches.

Required Privilege Level routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure)*
- Creating a Private VLAN on a Single QFX Switch on page 269
- Creating a Private VLAN Spanning Multiple QFX Series Switches on page 275

recovery-timeout

Syntax recovery-timeout *seconds*;

Hierarchy Level [edit interfaces *interface-name* unit 0 family ethernet-switching]

Release Information Statement introduced in Junos OS Release 15.1X49-D70.

Description Configure an interface to be temporarily disabled when MAC limiting is in effect with the action **shutdown**. This enables the affected interface to recover automatically from the error condition after the specified period of time:

- If you configure MAC limiting with the **shutdown** option and you enable **recovery-timeout**, the interface is temporarily disabled when the MAC address limit is reached. The interface will recover automatically after the number of seconds specified.

NOTE: The recovery-timeout configuration does not apply to preexisting error conditions. It impacts only error conditions that are detected after the recovery-timeout statement is configured and committed. To clear a preexisting error condition and restore the interface to service, use the operational mode commands clear ethernet-switching recovery-timeout .

Default The interface does not automatically recover from an error condition.

Options ***seconds***— Number of seconds that the interface remains in a disabled state due to a port error prior to automatic recovery.

Range: 10 through 3600

Required Privilege Level system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

Related Documentation

- clear ethernet-switching recovery-timeout on page 1172
- Understanding MAC Limiting on page 803
- Example: Configuring MAC Limiting on a Security Device on page 805
- Configuring Autorecovery From the Disabled State on Secure Interfaces of a Security Device (CLI Procedure) on page 808

redundancy-group (Interfaces)

Syntax **redundancy-group** *number* ;

Hierarchy Level **[edit interfaces** *interface-name* **redundant-ether-options]**

Release Information Statement introduced in Junos OS Release 9.0.

Description Specify the redundancy group that a redundant Ethernet interface belongs to.

Options ***number*** —Number of the redundancy group that the redundant interface belongs to. Failover properties of the interface are inherited from the redundancy group.

Range: 1 through 255

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Interfaces Feature Guide for Security Devices

redundant-trunk-group

Syntax

```
redundant-trunk-group {
  group name {
    interface interface-name <primary>;
    interface interface-name;
    preempt-cutover-timer seconds;
  }
}
```

Hierarchy Level

- For platforms with ELS:

  [edit switch-options]

- For platforms without ELS:

  [edit ethernet-switching-options]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Hierarchy level **[edit switch-options]** introduced in Junos OS Release 13.2X50-D10 (ELS). (See "Using the Enhanced Layer 2 Software CLI" on page 3 for information about ELS.)
Statement introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description

Configure a primary link and secondary link on trunk ports. If the primary link fails, the secondary link automatically takes over without waiting for normal spanning-tree protocol convergence.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support on page 613
- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

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.
Statement introduced in Junos OS Release 14.1X53-D35 for the EX 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 QFX Switches on page 846
- Configuring Reflective Relay on Switches on page 844

registration

Syntax registration (forbidden | normal | restricted);

Hierarchy Level [edit protocols mvrp interface (all | *interface-name*)],

[edit routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type),

[edit logical-systems *logical-system-name* protocols mvrp interface (all | *interface-name*)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp interface (all | *interface-name*)] (for virtual switch instance type)

Release Information Statement introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 10.1 for MX Series routers.
Statement introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description Specifies the Multiple VLAN Registration Protocol (MVRP) registration mode for the interface if MVRP is enabled.

Default **normal**—The interface or interfaces accept MVRP messages and participate in MVRP.

Options **forbidden**—The interface or interfaces do not register and do not participate in MVRP.

normal—The interface or interfaces accept MVRP messages and participate in MVRP.

restricted—The interface or interfaces ignore all MVRP JOIN messages received for VLANs that are not statically configured for MVRP on the interface.

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) on Switches on page 504
- *Example: Configuring Automatic VLAN Administration Using MVRP on MX Series Routers*
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Verifying That MVRP Is Working Correctly on page 551
- **join-timer (MVRP) on page 1010**
- **leaveall-timer (MVRP) on page 1022**
- **leave-timer (MVRP) on page 1020**
- **point-to-point on page 1065**

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 Switches on page 84
- *Understanding Routing Engines in the QFabric System*

restore-interval

Syntax restore-interval *number*;

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description Configures the number of minutes that the node does not process any Ethernet ring protection (ERP) protocol data units (PDUs).. This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value.

Options *number*—Specify the restore interval.

Range: 1 through 12 minutes

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

ring-protection-link-end

Syntax ring-protection-link-end;

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name* (east-interface | west-interface)]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description Specify that the port is one side of a ring protection link (RPL) by setting the RPL end flag.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

ring-protection-link-owner

Syntax ring-protection-link-owner;

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description Specify the ring protection link (RPL) owner flag in the Ethernet protection ring. Include this statement only once for each ring (only one node can function as the RPL owner).

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435

routing-instances

Syntax

```
routing-instances routing-instance-name {
  instance-type virtual-router;
  interface interface-name;
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 9.2 for EX Series switches.

Description Configure a virtual routing entity.

Options ***routing-instance-name***—Name for this routing instance.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.routing-control—To add this statement to the configuration.

Related Documentation

- Example: Using Virtual Routing Instances to Route Among VLANs on EX Series Switches on page 489
- Configuring Virtual Routing Instances on EX Series Switches (CLI Procedure) on page 488

secure-wire

Syntax secure-wire *secure-wire-name* interface [*interface-name-1 interface-name-2*];

Hierarchy Level [edit security forwarding-options]

Release Information Statement introduced in Junos OS Release 12.3X48-D10.

Description Configure mapping of interfaces through which traffic is forwarded unchanged.

Options **secure secure-wire**—Specify a name for the secure wire interface mapping.

interface-name-1 interface-name-2—Specify a pair of peer logical interfaces that constitutes the secure wire mapping.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- Understanding Secure Wire on Security Devices on page 679

security-zone

Syntax

```
security-zone zone-name {
  address-book {
    address address-name {
      ip-prefix {
        description text;
      }
      description text;
      dns-name domain-name {
        ipv4-only;
        ipv6-only;
      }
      range-address lower-limit to upper-limit;
      wildcard-address ipv4-address/wildcard-mask;
    }
    address-set address-set-name {
      address address-name;
      address-set address-set-name;
      description text;
    }
  }
  advance-policy-based-routing;
  application-tracking;
  description text;
  enable-reverse-reroute;
  host-inbound-traffic {
    protocols protocol-name {
      except;
    }
    system-services service-name {
      except;
    }
  }
  interfaces interface-name {
    host-inbound-traffic {
      protocols protocol-name {
        except;
      }
      system-services service-name {
        except;
      }
    }
  }
  screen screen-name;
  tcp-rst;
}
```

Hierarchy Level [edit security zones]

Release Information Statement introduced in Junos OS Release 8.5. Support for wildcard addresses added in Junos OS Release 11.1. The **description** option added in Junos OS Release 12.1.

Description Define a security zone, which allows you to divide the network into different segments and apply different security options to each segment.

Options ***zone-name*** —Name of the security zone.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation
- *Security Zones Overview*
- *Example: Configuring Application Firewall Rule Sets Within a Security Policy*

service-id

Syntax service-id *number*;

Hierarchy Level [edit switch-options]
[edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.
Statement introduced in Junos OS Release 13.2 for the QFX Series.

Description Specify a service identifier for each multichassis aggregated Ethernet interface that belongs to a link aggregation group (LAG).

Options **number**—A number that identifies a particular service.

Range: 1 through 65535

Required Privilege Level system—To view this statement in the configuration.
system control—To add this statement to the configuration.

shaping-rate (CoS Interfaces)

Syntax shaping-rate *rate* <overhead *bytes*> ;

Hierarchy Level [edit class-of-service interfaces *interface-name*],
[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]

Release Information Statement introduced in Junos OS Release 9.2.
overhead option introduced in Junos OS Release 18.1.

Description For logical interfaces on which you configure packet scheduling, configure traffic shaping by specifying the amount of bandwidth to be allocated to the logical interface.

Logical and physical interface traffic shaping can be configured together. This means you can include the **shaping-rate** statement at the **[edit class-of-service interfaces interface *interface-name*]** hierarchy level *and* the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number*]** hierarchy level. If you configure traffic shaping at both the logical and physical interface levels, the logical interface shaping credit is checked and updated before the physical interface shaping credit.

Alternatively, you can configure a shaping rate for a logical interface and oversubscribe the physical interface by including the **shaping-rate** statement at the **[edit class-of-service traffic-control-profiles]** hierarchy level. With this configuration approach, you can independently control the delay-buffer rate.

On the physical interface, you can set the Layer 2 overhead adjustment to the shaping rate calculation at egress.

Default If you do not include this statement at the **[edit class-of-service interfaces interface *interface-name* unit *logical-unit-number*]** hierarchy level, the default logical interface bandwidth is the average of unused bandwidth for the number of logical interfaces that require default bandwidth treatment. If you do not include this statement at the **[edit class-of-service interfaces interface *interface-name*]** hierarchy level, the default physical interface bandwidth is the average of unused bandwidth for the number of physical interfaces that require default bandwidth treatment.

Options **rate**—Peak rate, in bits per second (bps). You can specify a value in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation **k** (1000), **m** (1,000,000), or **g** (1,000,000,000).

Range: 1000 through 6,400,000,000,000 bps

overhead—Layer 2 shaping overhead adjustment to be applied at egress (bytes).

Range: -62 through 192

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *policer-overhead*

shutdown-threshold

Syntax shutdown-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 interface is disabled. Once an interface is disabled, you must explicitly reenable it using the **clear ethernet-switching layer2-protocol-tunneling error** command. Otherwise, the interface remains disabled.

The shutdown threshold value must be greater than or equal to the drop threshold value. If the shutdown threshold value is less than the drop threshold value, the drop threshold value has no effect.

You can specify a shutdown threshold value without specifying a drop threshold value.

Default No shutdown 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 interface is disabled.

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
- drop-threshold on page 919
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395

source-address (Security Policies)

Syntax

```
source-address {
    [address];
    any;
    any-ipv4;
    any-ipv6;
}
```

Hierarchy Level

[edit security policies from-zone *zone-name* to-zone *zone-name* policy *policy-name* match]

[edit security policies global policy *policy-name* match]

Release Information

Statement introduced in Junos OS Release 8.5. Support for IPv6 addresses added in Junos OS Release 10.2. Support for IPv6 addresses in active/active chassis cluster configurations (in addition to the existing support of active/passive chassis cluster configurations) added in Junos OS Release 10.4. Support for wildcard addresses added in Junos OS Release 11.1.

Description

Define the matching criteria. You can specify one or more IP addresses, address sets, or wildcard addresses. You can specify wildcards **any**, **any-ipv4**, or **any-ipv6**.

Options

address—IP addresses, address sets, or wildcard addresses (represented as A.B.C.D/wildcard-mask). You can configure multiple addresses or address prefixes separated by spaces and enclosed in square brackets.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Security Policies Overview*
- *Understanding Security Policy Rules*
- *Understanding Security Policy Elements*

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 Switches on page 84
- *Understanding Routing Engines in the QFabric System*

static-mac

Syntax

static-mac *mac-address*;

static-mac *mac-address* {
  vlan-id *number*;
}

Hierarchy Level

[edit vlans *vlan-name* switch-options interface *interface-name*]

[edit bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],

[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],

[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* bridge-options interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols evpn interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.4.
Statement modified in Junos OS Release 9.5.
Support for logical systems added in Junos OS Release 9.6.
[edit vlans *vlan-name* switch-options interface *interface name*] hierarchy level introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers. The **vlan-id** option is not available for EVPNs.
[edit vlans *vlan-name* switch-options interface *interface name*] hierarchy level introduced in Junos OS Release 13.2 for the QFX Series.

Description

Configure a static MAC address for a logical interface in a bridge domain or VLAN.

The **vlan-id** option can be specified for **static-macs** only if **vlan-id all** is configured for the bridging domain or VLAN.

Options

mac-address—MAC address

vlan-id *number*—(Optional) VLAN identifier to associate with static MAC address.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring EVPN Routing Instances*
- *Understanding Layer 2 Learning and Forwarding for Bridge Domains*
- Layer 2 Learning and Forwarding for VLANs Overview on page 27

- Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support (CLI Procedure) on page 56
- Understanding VLANs on Security Devices on page 380

swap

Syntax swap;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D20 for the QFX Series.

Description Specify the VLAN rewrite operation to replace a VLAN tag. The outer VLAN tag of the frame is overwritten with the user-specified VLAN tag information.

On MX Series routers, you can enter this statement on Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, aggregated Ethernet using Gigabit Ethernet IQ interfaces, and 100-Gigabit Ethernet Type 5 PIC with CFP. On EX Series switches, you can enter this statement on Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- *Rewriting the VLAN Tag on Tagged Frames*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

swap-push

Syntax swap-push;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Specify the VLAN rewrite operation to replace the outer VLAN tag of the frame with a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame.

You can use this statement on Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, and for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Rewriting a VLAN Tag and Adding a New Tag on page 208

swap-swap

Syntax swap-swap;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Specify the VLAN rewrite operation to replace both the inner and the outer VLAN tags of the frame with a user-specified VLAN tag value.

You can use this statement on Gigabit Ethernet IQ, IQ2 and IQ2-E interfaces, 10-Gigabit Ethernet LAN/WAN PIC, for aggregated Ethernet interfaces using Gigabit Ethernet IQ2 and IQ2-E or 10-Gigabit Ethernet PICs on MX Series routers, and for 100-Gigabit Ethernet Type 5 PIC with CFP.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Rewriting the Inner and Outer VLAN Tags*

switch-options (VLANs)

List of Syntax

Syntax (EX Series, MX Series, QFX Series and NFX Series) on page 1111
Syntax (SRX Series) on page 1111

Syntax (EX Series, MX Series, QFX Series and NFX Series)

```
switch-options {
  interface interface-name {
    interface-mac-limit limit {
      packet-action drop;
    }
    mac-pinning
    no-mac-learning;
    static-mac static-mac-address {
      vlan-id number;
    }
  }
  interface-mac-limit limit {
    packet-action drop;
  }
  mac-statistics;
  mac-ip-table-size number;
  mac-table-size limit {
    packet-action drop;
  }
  no-mac-learning;
  service-id number;
  vtep-source-interface
}
```

Syntax (SRX Series)

```
switch-options {
  interface interface-name {
    encapsulation-type;
    ignore-encapsulation-mismatch;
    pseudowire-status-tlv;
    static-mac mac-address {
      vlan-id vlan-id;
    }
  }
  mac-table-aging-time seconds;
  mac-table-size {
    number;
    packet-action drop;
  }
}
```

EX Series, MX Series, QFX Series and NFX Series

[edit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* vlans *vlan-name*],
[edit routing-instances *routing-instance-name* vlans *vlan-name*],
[edit vlans *vlan-name*]

SRX Series

[edit vlans *vlans-name*]

Release Information

Statement modified in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches and MX Series routers.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement (mac-pinning) introduced in Junos OS 16.2 for MX Series routers.
mac-ip-table-size statement introduced in Junos OS 17.4 Release for MX Series routers and EX9200 switches.

Description

Configure Layer 2 learning and forwarding properties for a VLAN or a virtual switch.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Ethernet Switching and Layer 2 Transparent Mode Overview on page 25

system-services (Security Zones Interfaces)

Syntax

```
system-services service-name {
  except;
}
```

Hierarchy Level [edit security zones security-zone *zone-name* interfaces *interface-name* host-inbound-traffic]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the types of traffic that can reach the device on a particular interface.

Options

- *service-name* —Service for which traffic is allowed. The following services are supported:
  - **all**—Enable all possible system services available on the Routing Engine (RE).
  - **any-service**—Enable services on entire port range.
  - **bootp**—Enable traffic destined to BOOTP and DHCP relay agents.
  - **dhcp**—Enable incoming DHCP requests.
  - **dhcpv6**—Enable incoming DHCP requests for IPv6.
  - **dns**—Enable incoming DNS services.
  - **finger**—Enable incoming finger traffic.
  - **ftp**—Enable incoming FTP traffic.
  - **http**—Enable incoming J-Web or clear-text Web authentication traffic.
  - **https**—Enable incoming J-Web or Web authentication traffic over Secure Sockets Layer (SSL).
  - **ident-reset**—Enable the access that has been blocked by an unacknowledged identification request.
  - **ike**—Enable Internet Key Exchange traffic.
  - **netconf SSH**—Enable incoming NetScreen Security Manager (NSM) traffic over SSH.
  - **ntp**—Enable incoming Network Time Protocol (NTP) traffic.
  - **ping**—Allow the device to respond to ICMP echo requests.
  - **r2cp**—Enable incoming Radio Router Control Protocol traffic.
  - **reverse-ssh**—Reverse SSH traffic.
  - **reverse-telnet**—Reverse Telnet traffic.
  - **rlogin**—Enable incoming **rlogin** (remote login) traffic.
  - **rpm**—Enable incoming real-time performance monitoring (RPM) traffic.
  - **rsh**—Enable incoming Remote Shell (**rsh**) traffic.

- **snmp**—Enable incoming SNMP traffic (UDP port 161).
- **snmp-trap**—Enable incoming SNMP traps (UDP port 162).
- **ssh**—Enable incoming SSH traffic.
- **telnet**—Enable incoming Telnet traffic.
- **tftp**—Enable TFTP services.
- **traceroute**—Enable incoming traceroute traffic (UDP port 33434).
- **xnm-clear-text**—Enable incoming Junos XML protocol traffic for all specified interfaces.
- **xnm-ssl**— Enable incoming Junos XML protocol-over-SSL traffic for all specified interfaces.

- **except**—(Optional) except can only be used if all has been defined.

Required Privilege Level

security—To view this statement in the configuration.
security-control—To add this statement to the configuration.

Related Documentation

- *Security Zones Overview*
- *Supported System Services for Host Inbound Traffic*

tag-protocol-id (TPIDs Expected to Be Sent or Received)

Syntax tag-protocol-id [*tpids*];

Hierarchy Level [edit interfaces *interface-name* gigether-options ethernet-switch-profile],
[edit interfaces *interface-name* aggregated-ether-options ethernet-switch-profile],
[edit interfaces *interface-name* aggregated-ether-options ethernet-switch-profile],
[edit interfaces *interface-name* ether-options ethernet-switch-profile]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 13.2X50-D15 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for the QFX Series.

Description For Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces, aggregated Ethernet with Gigabit Ethernet IQ interfaces, and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC, and the built-in Gigabit Ethernet port on the M7i router), define the TPIDs expected to be sent or received on a particular VLAN. For each Gigabit Ethernet port, you can configure up to eight TPIDs using the **tag-protocol-id** statement; but only the first four TPIDs are supported on IQ2 and IQ2-E interfaces.

For 10-Gigabit Ethernet LAN/WAN PIC interfaces on T Series routers only the default TPID value (**0x8100**) is supported.

For Gigabit Ethernet, 10-Gigabit Ethernet, 40-Gigabit Ethernet, and aggregated Ethernet interfaces on EX Series switches, define the TPIDs expected to be sent or received on a particular VLAN. The default TPID value is **0x8100**. Other supported values are **0x88a8**, **0x9100**, and **0x9200**.

Options ***tpids***—TPIDs to be accepted on the VLAN. Specify TPIDs in hexadecimal.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Frames with Particular TPIDs to Be Processed as Tagged Frames*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583

tag-protocol-id (TPID to Rewrite)

Syntax tag-protocol-id *tpid*;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description For Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2 and IQ2-E interfaces only, configure the outer TPID value. All TPIDs you include in input and output VLAN maps must be among those you specify at the **[edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]]** hierarchy level.

For 10-Gigabit Ethernet LAN/WAN PIC interfaces on T Series routers the default TPID value (**0x8100**) is supported.

Default If the **tag-protocol-id** statement is not configured, the TPID value is 0x8100.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Inner and Outer TPIDs and VLAN IDs*

traceoptions

List of Syntax

Ethernet Switching Options on page 1117
Ethernet Ring Protection on page 1117
Edge Virtual Bridging on page 1117

Ethernet Switching Options

```
traceoptions {
  file filename <files number> <no-stamp> <replace> <size size> <world-readable |
    no-world-readable>;
  flag flag <disable>;
}
```

Ethernet Ring Protection

```
traceoptions {
  file filename <no-stamp> <world-readable | no-world-readable> <replace> <size size>;
  flag flag;
}
```

Edge Virtual Bridging

```
traceoptions {
  file filename <files number> <no-stamp> <replace> <size size> <world-readable |
    no-world-readable>;
  flag flag;
}
```

Hierarchy Level

[edit ethernet-switching-options]

[edit protocols protection-group]

[edit protocols edge-virtual-bridging]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

NOTE: The **traceoptions** statement is not supported on the QFX3000 QFabric system.

Description

Define global tracing operations for access security features on Ethernet switches.

Configure trace options for the protection group.

Define global tracing operations for edge virtual bridging (EVB) features on Ethernet switches.

Default

The Ethernet Switching Options **traceoptions** feature is disabled by default.

Edge Virtual Bridging tracing operations are disabled by default.

Ethernet Ring Protection trace options are not set by default. On some EX Series switches, logging of basic ERPS state transitions is set by default. You can configure trace options on those switches to obtain more details than are provided by the default log. See "Understanding Ethernet Ring Protection Switching Functionality" on page 408 for additional information about default logging of the basic state transitions.

Options For Ethernet Switching 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.

For Ethernet Ring Protection:

file *filename*—Name of the file to receive the output of the tracing operation. All files are placed in the directory **/var/log**. You can include the following file options:

- **no-stamp**—(Optional) Do not timestamp trace file.
- **no-world-readable**—(Optional) Do not allow any user to read the log file.
- **replace**—(Optional) Replace the trace file rather than appending to it.
- **size**—(Optional) Maximum trace file size (10240..4294967295).
- **world-readable**—(Optional) Allow any user to read the log file.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. The following flags are available for both EX Switches and MX Series routers:

- **all**—Trace all
- **config** —Trace configuration messages
- **debug**—Trace debug messages
- **events**—Trace events to the protocol state machine
- **normal**—Trace normal messages
- **pdu** —Trace RAPS PDU reception and transmission
- **periodic-packet-management**—Trace periodic packet management state and events
- **state-machine**—Trace RAPS state machine
- **timers**—Trace protocol timers

For Edge Virtual Bridging:

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

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 output 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.

flag ***flag***—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—Trace everything.
- **ecp**—Trace Edge Control Protocol (ECP) events.
- **evb-tlv**— Trace EVB type, length, and value (TLV) events.
- **parse**—Trace configuration parsing.
- **policy**—Trace policy events.
- **vdp**—Trace Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP) events.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- *Overview of Spanning-Tree Protocols*
- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435
- Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure) on page 867

traceoptions (MVRP)

Syntax

```
traceoptions {
  file name <size size> <files number>  <(world-readable | no-world-readable)>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols mvrp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type),
[edit protocols mvrp],
[edit routing-instances *routing-instance-name* protocols mvrp] (for virtual switch instance type)

Release Information

Statement introduced in Junos OS Release 10.1 for MX Series routers.

Description

For Multiple VLAN Registration Protocol (MVRP), configure 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 *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. To include the **file** statement, you must specify a filename. Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place MVRP tracing output in the file **/var/log/mvrp-log**.

files *number*—(Optional) Maximum number of trace files, in the range from 2 through 1000. The default is 1 trace file. 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 **size** option.

flag *flag*—Specify which tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can include the following flags:

- **all**—Enable all trace options flags.
- **error**—Trace all failure conditions.
- **events**—Trace process state change and cleanup events.
- **pdu**—Trace RAPS PDU reception and transmission.
- **socket**—Trace socket activity.

- **state-machine**—Trace information about the state machine.
- **timers**—Trace protocol timers.

no-world-readable—(Optional) Prevent any user from reading the log 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. The file size range is from 10240 through 4294967295. The default file size is 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 Automatic VLAN Administration Using MVRP on MX Series Routers*
- Configuring Multiple VLAN Registration Protocol (MVRP) to Manage Dynamic VLAN Registration on page 511
- Verifying That MVRP Is Working Correctly on page 551
- Understanding Multiple VLAN Registration Protocol (MVRP) for Dynamic VLAN Registration on page 501

unconditional-src-learn

Syntax unconditional-src-learn;

Hierarchy Level [edit interfaces *interface-name* unit *logical-unit-number* family inet]

Release Information Statement introduced in Junos OS Release 10.4R16.

Description Enables the router to learn IP addresses from nonvalidated sources when proxy Address Resolution Protocol (ARP) is configured.

CAUTION: By default, the router learns IP addresses from validated sources only. When this statement is configured and proxy ARP is enabled on an unnumbered interface, the router responds to ARP requests from any IP address, which might lead to exploitable information disclosure. An attacker can poison the ARP cache and create a fake forwarding table entry for an IP address, effectively creating a denial of service for that subscriber or interface. Therefore, exercise caution when configuring this statement.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Understanding Proxy ARP on page 625
- Example: Configuring Proxy ARP on an EX Series Switch on page 629

unframed | no-unframed (Interfaces)

Syntax (unframed | no-unframed);

Hierarchy Level [edit interfaces *interface-name* t3-options]

Release Information Statement introduced in Junos OS Release 11.1.

Description Enable or disable framing for the T3 interface on a 1-Port Clear Channel DS3/E3 GPIM on an SRX Series device. By default, unframed mode is enabled. Select no-unframed to enable framing. Select unframed to return to the default mode.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Example: Configuring a T3 Interface*

unicast-in-lpm

Syntax unicast-in-lpm;

Hierarchy Level [edit chassis forwarding-options lpm-profile]

Release Information Statement introduced in Junos OS Release 14.1x53-D30 for QFX Series switches.

Description For the Unified Forwarding Table feature, specify to store all unicast IPv4 and IPv6 entries with prefixes with lengths equal to or less than 64 in the table for longest prefix match (LPM) entries, thereby freeing up space in the Layer 3 host table. Only unicast entries can be moved to the LPM table. Multicast entries must be stored in the Layer 3 host table.

You can also configure this statement in conjunction with the prefix-65-127-disable statement, which allocates no memory for IPv6 prefixes with lengths in the range /65 through /127. Together, these two statements allocate more space for unicast IPv4 and IPv6 entries with prefix lengths equal to or less than 64.

NOTE: This statement is supported only on the lpm-profile.

This statement is not supported on QFX5200 switches.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Understanding the Unified Forwarding Table*

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. See CLI Explorer.

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

- show ethernet-switching table on page 1243
- show vlans on page 1510

vlan

List of Syntax

Syntax (Ethernet and 802.1Q Tagging) on page 1128
Syntax (Static MAC-based VLANs) on page 1128
Syntax (Unknown Unicast) on page 1128

Syntax (Ethernet and 802.1Q Tagging)

```
vlan {
  members [ (all | names | vlan-ids) ];
}
```

Syntax (Static MAC-based VLANs)

```
vlan vlan-name {
  mac mac-address {
      next-hop interface-name;
  }
}
```

Syntax (Unknown Unicast)

```
vlan (all | vlan-name) {
  interface interface-name;
}
```

Ethernet

[edit interfaces ge-*chassis/slot/port* unit *logical-unit-number* ethernet-switching],
[edit interfaces xe-*chassis/slot/port* unit *logical-unit-number* ethernet-switching]

802.1Q Tagging

[edit interfaces *interface-name* unit *logical-unit-number* family ethernet-switching]

Static MAC-based VLANs

[edit ethernet-switching-options static]

Unknown Unicast

[edit ethernet-switching-options unknown-unicast-forwarding]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description

For both Gigabit Ethernet and aggregated Ethernet interfaces and 802.1Q Tagging, assign an 802.1Q VLAN tag ID to a logical interface.

For Static MAC-based VLANs, specify the name of a VLAN to add to the Ethernet switching table.

For unknown unicast, 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 statements are explained separately. See CLI Explorer.

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 the VLAN to add to the Ethernet switching table.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
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 141
- Adding a Static MAC Address Entry to the Ethernet Switching Table (CLI Procedure) on page 55
- **show ethernet-switching interfaces on page 1215**
- **show ethernet-switching table on page 1243**
- **show ethernet-switching interface on page 1212**
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Configuring Routed VLAN Interfaces on Switches (CLI Procedure) on page 365
- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) on page 456
- Understanding Bridging and VLANs on Switches on page 84
- **show vlans on page 1510**
- *Junos OS Network Interfaces Configuration Guide*

vlan-id

Syntax vlan-id (all | none | *number*);

VLANs and Bridge Domain VLANs

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]

[edit vlans *vlan-name*],
[edit logical-systems *logical-system-name* vlans*vlan-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* vlans*vlan-name*],
[edit routing-instances *routing-instance-name* vlans *vlan--name*]

802.1Q Tagging [edit vlans *vlan-name*]

VLAN ID to Rewrite

[edit interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* input-vlan-map],
[edit logical-systems *logical-system-name* interfaces *interface-name* unit *logical-unit-number* output-vlan-map]

VLAN Tagging and Layer 3 Subinterfaces [edit interfaces *interface-name* unit *logical-unit-number*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.2 for EX Series switches VLAN tagging and Layer 3 subinterfaces.
Support for Layer 2 trunk ports added in Junos OS Release 9.2.
Support for SRX 5600, and SRX 5800 devices added in Junos OS Release 9.6.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description For VLANs, specify a VLAN identifier (VID) to include in the packets sent to and from the VLAN, or a VPLS routing instance.

NOTE: When configuring a VLAN identifier for provider backbone bridge (PBB) routing instances, dual-tagged VIDs and the none option are not permitted.

For 802.1Q tagging, configure an 802.1Q tag to apply to all traffic that originates on the VLAN.

The number zero is reserved for priority tagging and the number 4095 is also reserved.

For VLAN ID to Rewrite Gigabit Ethernet IQ and 10-Gigabit Ethernet IQ2, 10-Gigabit Ethernet LAN/WAN PIC, and IQ2-E interfaces and aggregated Ethernet using Gigabit Ethernet IQ interfaces, specify the line VLAN identifiers to be rewritten at the input or output interface.

You cannot include the **vlan-id** statement with the **swap** statement, **swap-push** statement, **push-push** statement, or **push-swap** statement at the **[edit interfaces** ***interface-name*** **unit** ***logical-unit-number*** output-vlan-map**]** hierarchy level. If you include any of those statements in the output VLAN map, the VLAN ID in the outgoing frame is rewritten to the **vlan-id** statement that you include at the **[edit interfaces** ***interface-name*** **unit** ***logical-unit-number*****]** hierarchy level.

Default For 802.1Q Tagging on EX Series and SRX Series, If you use the default factory configuration, all traffic originating on the VLAN is untagged and has a VLAN identifier of **1**.

For VLANs 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.

For VLAN tagging and Layer 3 subinterfaces, bind an 802.1Q VLAN tag ID to a logical interface.

NOTE: The VLAN tag ID cannot be configured on logical interface unit **0**. The logical unit number must be 1 or higher.

Options For VLANs:

number—A valid VLAN identifier. If you configure multiple VLANs with a valid VLAN identifier, you must specify a unique VLAN identifier for each. However, you can use the same VLAN identifier for VLANs that belong to different virtual switches. Use this option to send single tagged frames with the specified VLAN identifier over VPLS VT interfaces.

NOTE: If you specify a VLAN identifier, you cannot also use the all option. They are mutually exclusive.

all—Specify that the VLAN spans all the VLAN identifiers configured on the member logical interfaces.

NOTE: You cannot specify the all option if you include a routing interface in the VLAN.

none—Specify to enable shared VLAN learning or to send untagged frames over VPLS VT interfaces.

NOTE: Multichassis link aggregation (MC-LAG) does not support the none option with the vlan-id statement with VLANs.

For 802.1Q Tagging:

number —VLAN tag identifier

Range:

- 1 through 4094 (all switches except EX8200 Virtual Chassis)
- 1 through 4092 (EX8200 Virtual Chassis only)

Default: 1

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.
system—To view this statement in the configuration.
system-control—To add this statement to the configuration.
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Example: Connecting Access Switches with ELS Support to a Distribution Switch with ELS Support on page 170
- Example: Configuring a Private VLAN on a Single Switch with ELS Support on page 291
- Creating a Private VLAN on a Single Switch with ELS Support (CLI Procedure) on page 273
- Creating a Private VLAN Spanning Multiple EX Series Switches (CLI Procedure) on page 277
- Example: Configuring VLANs on Security Devices on page 382
- *Example: Configuring Interfaces and Routing Instances for a User Logical Systems*
- *Rewriting the VLAN Tag on Tagged Frames*
- *Binding VLAN IDs to Logical Interfaces*
- **vlan-tagging on page 1142**
- *Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure)*
- *Configuring Gigabit Ethernet Interfaces (CLI Procedure) for EX Series Switches with ELS support*
- *Configuring Gigabit Ethernet Interfaces (J-Web Procedure)*
- *Configuring a Layer 3 Subinterface (CLI Procedure)*
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- *Junos OS Ethernet Interfaces Configuration Guide*

vlan-id-list

Syntax vlan-id-list [*vlan-id-numbers*];

Hierarchy Level [edit bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit interfaces *interface-name* unit 0],
[edit interfaces *interface-name* unit *logical-unit-number*],
[edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.4.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.

Description Specify a VLAN identifier list to use for a bridge domain or VLAN in trunk mode.

Specify the **trunk** option in the **interface-mode** statement to accept packets with a VLAN ID that matches the list of VLAN IDs specified in the **vlan-id-list** statement to forward the packet within the bridge domain or VLAN configured with the matching VLAN ID. Specify the **access** option to accept packets with no VLAN ID to forward the packet within the bridge domain or VLAN configured with the VLAN ID that matches the VLAN ID specified in the **vlan-id** statement.

This statement also enables you to bind a logical interface to a list of VLAN IDs, thereby configuring the logical interface to receive and forward a frame with a tag that matches the specified VLAN ID list.

WARNING: On some EX and QFX Series switches, you can apply no more than eight VLAN identifier lists to a physical interface.

Options ***vlan-id-numbers***—Valid VLAN identifiers. You can combine individual numbers with range lists by including a hyphen.

Range: 0 through 4095

NOTE: On EX Series switches and the QFX Series, the range is 0 through 4094.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring a Bridge Domain*
- Configuring a VLAN on page 202
- *Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances*
- Configuring VLAN Identifiers for VLANs and VPLS Routing Instances on page 371

vlan-id-range

Syntax vlan-id-range *vlan-id–vlan-id*

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 in Junos OS Release 8.4.

Description Bind a range of VLAN IDs to a logical interface.

Options **number**—The first number is the lowest VLAN ID in the range the second number is the highest VLAN ID in the range.

Range: 1 through 4094

NOTE: On SRX240, SRX300, SRX320, SRX340, SRX345, SRX550M, and SRX650 devices, the VLAN range from 1 to 4094 on inet interfaces and the VLAN range from 1 to 3967 on Ethernet switching interfaces. On Ethernet switching interfaces, the VLAN range from 3968 to 4094 falls under the reserved VLAN address range, and the user is not allowed to configure VLANs in this range.

NOTE: Configuring **vlan-id-range** with the entire vlan-id range is an unnecessary waste of system resources and is not best practice. It should be used only when a subset of VLAN IDs (not the entire range) needs to be associated with a logical interface. If you specify the entire range (1-4094), it has the same result as not specifying a range; however, it consumes PFE resources such as VLAN lookup tables entries, and so on.

The following examples illustrate this further:

```
[edit interfaces interface-name]
vlan-tagging;
unit number {
    vlan-id-range 1-4094;
}

[edit interfaces interface-name]
unit 0;
```

VLAN ID 0 is reserved for tagging the priority of frames.

Required Privilege Level
interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation
- Understanding VLANs on page 769

vlan-id-start

Syntax
vlan-id-start *S-VLAN-ID*;

Hierarchy Level
[edit vlans *vlan-name* interface *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 QFX Series Switches on page 581
- Example: Setting Up Q-in-Q Tunneling on QFX Series Switches on page 598

vlan-prune

Syntax vlan-prune;

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement introduced in Junos OS Release 12.3 for EX Series switches.

Description Prune the Virtual Chassis port (VCP) paths in a Virtual Chassis to ensure received broadcast, multicast, and unknown unicast traffic in a VLAN uses the shortest possible path through the Virtual Chassis to the egress VLAN interface.

By default, all broadcast, multicast, and unknown unicast traffic in a VLAN on an EX Series Virtual Chassis is broadcast to all member switches in the Virtual Chassis. This behavior unnecessarily consumes bandwidth within the Virtual Chassis because unneeded traffic is sent to all Virtual Chassis member switches.

Enabling this option allows you to conserve bandwidth within the Virtual Chassis. Broadcast, multicast, and unknown unicast traffic still enters and exits the Virtual Chassis within the same VLAN, without the added bandwidth consumption that results from broadcasting this traffic to all member switches.

Default Disabled

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98

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 9.2 for EX Series switches.
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 system—To view this statement in the configuration.
system–control—To add this statement to the configuration.
routing—To view this statement in the configuration.
routing–control—To add this statement to the configuration.

Related Documentation

- Configuring VLANs on Switches on page 93
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98
- *Configuring VLANs for EX Series Switches (J-Web Procedure)*
- Configuring Routed VLAN Interfaces on Switches (CLI Procedure) on page 365
- Understanding Bridging and VLANs on Switches on page 84
- Configuring IRB Interfaces on Switches on page 454

vlan-rewrite

Syntax vlan-rewrite translate (200 500 | 201 501)

Hierarchy Level [edit interfaces *interface-name* unit *number* family bridge interface-mode trunk]
[edit interfaces *interface-name* unit *number* family ethernet-switching interface-mode trunk]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description Translates an incoming VLAN to a bridge-domain VLAN, corresponding counter translation at egress. Supports translation of VLAN 200 to VLAN 500 and VLAN 201 to VLAN 501. Other valid VLANs pass through without translation.

Options

translate 200 500—Translates incoming packets with VLAN 200 to 500.

translate 201 501—Translates incoming packets with VLAN 201 to 501.

translate 202 502—Translates incoming packets with VLAN 202 to 502.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Rewriting a VLAN Tag and Adding a New Tag on page 208

vlan-tagging

Syntax vlan-tagging;

Syntax (QFX Series, NFX Series, and EX4600) vlan-tagging;

Syntax (SRX Series Interfaces) vlan-tagging native-vlan-id *vlan-id*;

Hierarchy Level [edit interfaces *interface-name*],
[edit logical-systems *logical-system-name* interfaces *interface-name*]

QFX Series, NFX Series, and EX4600 Interfaces [edit interfaces (QFX Series) *interface-name*]
[edit interfaces (QFX Series) interface-range *interface-range-name*]

SRX Series Interfaces [edit interfaces *interface*]

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 9.5.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.2 for ACX Series Universal Metro Routers.
Statement introduced in Junos OS Release 13.2 for PTX Series Routers.
Statement introduced in Junos OS Release 14.1X53-D10 for the QFX Series.

Description For Fast Ethernet and Gigabit Ethernet interfaces, aggregated Ethernet interfaces configured for VPLS, and pseudowire subscriber interfaces, enable the reception and transmission of 802.1Q VLAN-tagged frames on the interface.

NOTE: For QFX Series configure VLAN identifier for untagged packets received on the physical interface of a trunk mode interface. Enable VLAN tagging. The platform receives and forwards single-tag frames with 802.1Q VLAN tags.

On EX Series switches except for EX4300 and EX9200 switches, the **vlan-tagging** and **family ethernet-switching** statements cannot be configured on the same interface. Interfaces on EX2200, EX3200, EX3300, EX4200, and EX4500 switches are set to **family ethernet-switching** by the default factory configuration. EX6200 and EX8200 switch interfaces do not have a default **family** setting.

Default VLAN tagging is disabled by default.

Options

native-vlan-id— (SRX Series)Configures a VLAN identifier for untagged packets. Enter a number from 0 through 4094.

NOTE: The **native-vlan-id** can be configured only when either **flexible-vlan-tagging** mode or **interface-mode** trunk is configured.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *802.1Q VLANs Overview*
- *Configuring a Layer 3 Subinterface (CLI Procedure)*
- *Configuring Tagged Aggregated Ethernet Interfaces*
- *Example: Configuring Layer 3 Subinterfaces for a Distribution Switch and an Access Switch*
- *vlan-id*
- *Configuring a Layer 3 Logical Interface*
- *Configuring VLAN Tagging*

vlan-tags

Syntax vlan-tags outer *number* inner *number*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]
[edit vlans *vlan-name*]

Release Information
Statement introduced in Junos OS Release 8.4.
Support for logical systems added in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.
Statement introduced in Junos OS Release 13.2X51-D10 for QFX Series switches.

Description Specify dual VLAN identifier tags for a bridge domain, VLAN, or VPLS routing instance.

Options
outer *number*—A valid VLAN identifier.

inner *number*—A valid VLAN identifier.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Configuring a Bridge Domain*
- Configuring a VLAN on page 202
- *Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances*
- Configuring VLAN Identifiers for VLANs and VPLS Routing Instances on page 371
- *Configuring a Layer 2 Virtual Switch* .
- Configuring a Layer 2 Virtual Switch on an EX Series Switch on page 95

vlan members (VLANs)

Syntax vlan members [*vlan-id*];

Hierarchy Level [edit vlans *vlan-name*]

Release Information Statement modified in Junos OS Release 9.5.

Description Specify multiple VLAN identifiers to create a VLAN for each VLAN identifier.

Options ***vlan-id***—A list of valid VLAN identifiers. A VLAN is created for each VLAN identifier in the list.

NOTE: If you specify a VLAN identifier list, you cannot configure an IRB interface in 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 VLANs on Security Devices on page 382

vlans

List of Syntax

Syntax (QFX Series, QFabric, NFX Series and EX4600)

```
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;
    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;
  }
}
```

Syntax (QFX Series, NFX Series and EX4600)

```
vlans {
  vlan-name {
    description text-description;
    domain-type bridge;
    forwarding-options {
      dhcp-security {
        arp-inspection;
        group group-name {
          interface interface-name {
            static-ip ip-address {
              mac mac-address;
            }
          }
          overrides {
            no-option82;
            trusted;
            untrusted;
          }
        }
        ip-source-guard;
```

```
    no-dhcp-snooping;
    option-82 {
      circuit-id {
        prefix {
          host-name;
          logical-system-name;
          routing-instance-name;
        }
        use-interface-description (device | logical);
        use-vlan-id;
      }
      remote-id {
        host-name hostname;
        use-interface-description (device | logical);
        use-string string;
      }
      vendor-id {
        use-string string;
      }
    }
  }
  fip-security {
    examine-vn2vf;
    examine-vn2vn {
      beacon-period milliseconds;
    }
    fc-map fc-map-value;
    interface interface-name {
      (fcoe-trusted | no-fcoe-trusted;)
    }
  }
}
l3-interface irb.logical-unit-number;
multicast-snooping-options {
  flood-groups [group-names];
  forwarding-cache {
    threshold {
      reuse threshold;
      suppress threshold;
    }
  }
  graceful-restart {
    disable;
    restart-duration duration;
  }
  host-outbound-traffic {
    dot1p bits;
    forwarding-class forwarding-class;
  }
  multichassis-lag-replicate-state;
  nexthop-hold-time time;
  options {
    syslog {
      level level;
      mark interval;
      upto level;
```

```
        }
      }
      traceoptions {
        file filename {
          files number;
          no-world-readable;
          size file-size;
          world-readable;
        }
        flag flag {
          disable;
        }
      }
    }
    switch-options {
      interface interface-name {
        interface-mac-limit limit {
          packet-action action;
        }
        static-mac mac-address;
      }
      interface-mac-limit limit {
        packet-action action;
      }
      mac-move-limit limit {
        packet-action action;
      }
      mac-table-size limit {
        packet-action drop;
      }
      no-mac-learning;
      }
    }
    vlan-id number;
    vlan-id-list [vlan-id | vlan-id–vlan-id];
    vlan-tags
      inner value;
      outer value;
    }
    vxlan {
      ingress-node-replication
      ovsdb-managed
      }
    }
  }
}
```

Syntax (SRX Series and EX Series)

```
vlans {
  vlan-name {
    description text-description;
    dot1q-tunneling {
      customer-vlans (id | range)
      layer2-protocol-tunneling all | protocol-name {
        drop-threshold number;
        shutdown-threshold number;
      }
    }
    filter input filter-name;
    filter output filter-name;
    interface interface-name {
      egress;
      ingress;
      mapping (native (push | swap) | policy | tag (push | swap));
      pvlan-trunk;
    }
    isolation-id id-number;
    l3-interface l3-interface-name.logical-interface-number;
    l3-interface-ingress-counting layer-3-interface-name;
    mac-limit limit action action;
    mac-table-aging-time seconds;
    no-local-switching;
    no-mac-learning;
    primary-vlan vlan-name;
    vlan-id number;
    vlan-prune;
    vlan-range vlan-id-low-vlan-id-high;
  }
}
```

Syntax (SRX Series)

```
vlans {
  vlan name {
    (vlan-id (1..3967) | vlan-id-list [ vlan-id-numbers ]);
    description;
    forwarding-options {
      dhcp-security {
        arp-inspection;
        dhcpv6-options {
          option-16 {
            use-string use-string;
          }
          option-18 {
            prefix {
              host-name;
              logical-system-name;
              routing-instance-name;
              vlan-id;
              vlan-name;
            }
            use-interface-description (device | logical);
            use-interface-index (device | logical);
            use-interface-mac;
            use-interface-name (device | logical);
            use-string use-string;
          }
          option-37 {
            prefix {
              host-name;
              logical-system-name;
              routing-instance-name;
              vlan-id;
              vlan-name;
            }
            use-interface-description (device | logical);
            use-interface-index (device | logical);
            use-interface-mac;
            use-interface-name (device | logical);
            use-string use-string;
          }
        }
        group group-name {
          interface interface-name {
            static-ip {
              ip-address {
              mac-address;
              }
            }
            static-ipv6 {
              ip-address {
              mac-address;
              }
            }
          }
          overrides {
            no-dhcpv6-options;
            no-option16;
```

```
            no-option18;
            no-option37;
            no-option82;
            trusted;
            untrusted;
          }
        }
        ip-source-guard;
        ipv6-source-guard;
        neighbor-discovery-inspection;
        no-dhcp-snooping;
        no-dhcpv6-snooping;
        option-82 {
          circuit-id {
            prefix {
              host-name;
              logical-system-name;
              routing-instance-name;
            }
            use-interface-description (device | logical);
            use-vlan-id;
          }
          remote-id {
            host-name;
            mac;
            use-interface-description (device | logical);
            use-string use-string;
          }
          vendor-id {
            use-string use-string;
          }
        }
      }
      filter {
        input filter-name;
      }
      flood {
        input filter-name;
      }
    }
    interface interface-name;
    l3-interface l3-interface-name;
    mcae-mac-flush;
    mcae-mac-synchronize;
    service-id service-id;
    switch-options {
      interface name {
        action-priority action-priority;
        encapsulation-type (ethernet | ethernet-vlan);
      ignore-encapsulation-mismatch;
      interface-mac-limit {
        limit;
        packet-action (drop | drop-and-log | log | none | shutdown);
      }
      no-mac-learning;
      pseudowire-status-tlv;
```

```
            static-mac mac-address {
              vlan-id value;
            }
            }
            interface-mac-limit {
              limit;
              packet-action (drop | drop-and-log | log | none | shutdown);
            }
            mac-table-aging-time seconds;
            mac-table-size {
              limit;
              packet-action {
              drop;
              }
            }
            no-mac-learning;
            static-rvtep-mac {
              mac mac_addr {
              remote-vtep;
            }
            }
          }
        }
      }
```

Syntax (vSRX)

```
vlans {
  vlan name {
    (vlan-id (all | none | number) | vlan-id-list [ vlan-id-numbers ] | vlan-tags <inner number>
      outer number);
    description;
    forwarding-options {
      dhcp-security {
        arp-inspection;
        dhcpv6-options {
          option-16 {
            use-string use-string;
          }
          option-18 {
            prefix {
              host-name;
              logical-system-name;
              routing-instance-name;
              vlan-id;
              vlan-name;
            }
            use-interface-description (device | logical);
            use-interface-index (device | logical);
            use-interface-mac;
            use-interface-name (device | logical);
            use-string use-string;
          }
          option-37 {
            prefix {
              host-name;
              logical-system-name;
              routing-instance-name;
              vlan-id;
              vlan-name;
            }
            use-interface-description (device | logical);
            use-interface-index (device | logical);
            use-interface-mac;
            use-interface-name (device | logical);
            use-string use-string;
          }
        }
        group group-name {
          interface interface-name {
            static-ip {
              ip-address;
            }
            static-ipv6 {
              ip-address;
            }
          }
          overrides {
            no-dhcpv6-options;
            no-option16;
            no-option18;
            no-option37;
            no-option82;
```

```
            trusted;
            untrusted;
          }
        }
        ip-source-guard;
        ipv6-source-guard;
        light-weight-dhcpv6-relay;
        neighbor-discovery-inspection;
        no-dhcp-snooping;
        no-dhcpv6-snooping;
        option-82 {
          circuit-id {
            prefix {
              host-name;
              logical-system-name;
              routing-instance-name;
            }
            use-interface-description (device | logical);
            use-vlan-id;
          }
          remote-id {
            host-name;
            mac;
            use-interface-description (device | logical);
            use-string use-string;
          }
          vendor-id {
            use-string use-string;
          }
        }
      }
      filter {
        input filter-name;
      }
      flood {
        input filter-name;
      }
    }
    interface interface-name;
    l3-interface l3-interface-name;
    mcae-mac-synchronize;
    no-irb-layer-2-copy;
    service-id service-id;
    switch-options {
      interface name {
        action-priority action-priority;
        encapsulation-type (ethernet | ethernet-vlan);
      ignore-encapsulation-mismatch;
      interface-mac-limit {
        disable;
        limit;
        packet-action (drop | drop-and-log | log | none | shutdown);
      }
      mac-pinning;
      no-mac-learning;
      pseudowire-status-tlv;
```

```
        static-mac mac-address {
          vlan-id value;
        }
        }
        interface-mac-limit {
          limit;
          packet-action (drop | drop-and-log | log | none | shutdown);
        }
        mac-statistics;
        mac-table-aging-time seconds;
        mac-table-size {
          limit;
          packet-action {
          drop;
          }
        }
        no-mac-learning;
        static-rvtep-mac {
          mac mac_addr {
          remote-vtep;
        }
        }
      }
    }
}
```

Hierarchy Level

[edit]

[edit routing-instances *routing-instance-name*]

Release Information

Statement introduced in Junos OS Release 9.0 for EX Series switches.
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.
Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 15.1X49-D10.

Description

Configure VLAN properties.

On EX Series switches and SRX Series devices (including vSRX), the following configuration guidelines apply:

- Only private VLAN (PVLAN) firewall filters can be used when the VLAN is enabled for Q-in-Q tunneling.
- An S-VLAN tag is added to the packet if the VLAN is Q-in-Q-tunneled and the packet is arriving from an access interface.
- You cannot use a firewall filter to assign an integrated routing and bridging (IRB) interface or a routed VLAN interface (RVI) to a VLAN.
- VLAN assignments performed using a firewall filter override all other VLAN assignments.

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 include letters, numbers, hyphens (-), and periods (.) and can contain up to 255 characters long.

The remaining statements are explained separately. See CLI Explorer.

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.
system—To view this statement in the configuration.
system–control—To add this statement to the configuration.

Related Documentation

- Configuring IRB Interfaces on Switches on page 454
- Understanding Bridging and VLANs on Switches on page 84
- Configuring VLANs on Switches with Enhanced Layer 2 Support on page 97
- Configuring VLANs for EX Series Switches (CLI Procedure) on page 98
- Configuring VLANs for EX Series Switches with ELS Support (CLI Procedure) on page 102
- Example: Configuring VLANs on Security Devices (CLI Procedure) on page 771
- Configuring Q-in-Q Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 583
- Configuring Q-in-Q Tunneling on Security Devices on page 573
- Configuring Integrated Routing and Bridging Interfaces on Switches (CLI Procedure) on page 456
- Understanding Bridging and VLANs on Switches on page 84

vrf-mtu-check

Syntax vrf-mtu-check;

Hierarchy Level [edit chassis]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description On M Series routers (except the M120 and M320 router), T Series routers, and on EX Series 8200 switches, configure path maximum transmission unit (MTU) checks on the outgoing interface for unicast traffic routed on a virtual private network (VPN) routing and forwarding (VRF) instance.

Default Disabled.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *Configuring Path MTU Checks for VPN Routing Instances*
- *Configuring the Junos OS to Enable MTU Path Check for a Routing Instance on M Series Routers*

vsi-discovery

Syntax

```
vsi-discovery {
  interface interface-name
  vsi-policy vsi-policy-name
}
```

Hierarchy Level [edit protocols edge-virtual-bridging]

Description Configure Virtual Station Interface (VSI) Discovery and Configuration Protocol (VDP). VDP is used to program policies for each individual station interface (VSI).

Default VDP is disabled by default.

Options **interface-name**—Name of the interface on which VDP is configured. 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

- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859
- Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure) on page 867

vsi-policy

Syntax vsi-policy *vsi-policy-name*from vsi-manager *vsi-manager-id* vsi-type *vsi-type* vsi-version *vsi-version* vsi-instance *instance-number*;

Hierarchy Level [edit policy-options]

Description Define and apply the named VSI policy to the edge virtual bridging (EVB) configuration. For use with edge virtual bridging, each virtual machine (VM) on the server is uniquely identified by following four parameters, which are contained in a VSI policy:

- vsi-manager-id
- vsi-type
- vsi-version
- vsi-instance-id

The vsi-policy command manually configures these four parameters on the EX switch for the successful association of VM-VSI. VDP protocol helps determine the parameters defined for the virtual machines on the server and configure them on the switch. Use policy options to define the VM-VSI parameters. Configure a firewall filter for each of the VM profiles and use it in this statement.

Required Privilege Level system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859
- Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure) on page 867

west-interface

Syntax

```
west-interface {
  node-id mac-address;
  control-channel channel-name {
    vlan number;
    interface name interface-name
  }
  interface-none
  ring-protection-link-end;
  virtual-control-channel {
    west-interface name;
    east-interface name;
  }
}
```

Hierarchy Level [edit protocols protection-group ethernet-ring *ring-name*]

Release Information

Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.153-D10 for QFX Series switches.

Description Define one of the two interface ports for Ethernet ring protection, the other being defined by the **east-interface** statement at the same hierarchy level. The interface must use the control channel's logical interface name. The control channel is a dedicated VLAN channel for the ring port.

NOTE: Always configure this port second, after configuring the east-interface statement.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Ethernet Ring Protection Switching Overview on page 407
- *Ethernet Ring Protection Using Ring Instances for Load Balancing*
- **east-interface on page 921**
- **ethernet-ring on page 934**
- Example: Configuring Ethernet Ring Protection Switching on EX Series Switches on page 420
- Example: Configuring Ethernet Ring Protection Switching on QFX Series and EX Series Switches Supporting ELS on page 435

- Configuring Ethernet Ring Protection Switching on Switches (CLI Procedure) on page 416

CHAPTER 43

Operational Commands

- show ethernet-switching layer2-protocol-tunneling vlan
- show ethernet-switching mac-learning-log
- show ethernet-switching statistics
- show ethernet-switching statistics aging
- show ethernet-switching statistics mac-learning
- show ethernet-switching table
- show interfaces
- show interfaces irb
- show interfaces queue
- show interfaces swfabx
- show mac-rewrite interface
- show mvrp
- show mvrp applicant-state
- show mvrp dynamic-vlan-memberships
- show mvrp interface
- show mvrp registration-state
- show mvrp statistics
- show oam ethernet connectivity-fault-management adjacencies
- show oam ethernet connectivity-fault-management forwarding-state
- show oam ethernet connectivity-fault-management interfaces
- show oam ethernet connectivity-fault-management mep-database
- show oam ethernet connectivity-fault-management mep-statistics
- show oam ethernet connectivity-fault-management mip
- show oam ethernet connectivity-fault-management path-database
- show oam ethernet link-fault-management
- show protection-group ethernet-ring aps
- show protection-group ethernet-ring configuration
- show protection-group ethernet-ring data-channel
- show protection-group ethernet-ring interface
- show protection-group ethernet-ring node-state
- show protection-group ethernet-ring statistics
- show protection-group ethernet-ring vlan
- show redundant-trunk-group
- show security flow gate family
- show security flow ip-action
- show security flow session family
- show security flow statistics

- show security flow status
- show security forward-options secure-wire
- show security policies
- show security zones
- show system statistics arp
- show vlans
- traceroute ethernet

clear dot1x

Syntax

clear dot1x
interface <*interface-name*>
mac-address <*static-mac-address*>
statistics <interface *interface-name*>

Release Information

Command introduced in Junos OS Release 15.1X49-D80.

Description

Reset the authentication state of an interface or delete 802.1X statistics from the device. When you reset an interface using the **interface** or **mac-address** options, reauthentication on the interface is also triggered. The device sends out a multicast message on the interface to restart the authentication of all connected supplicants. If a MAC address is reset, then the device sends out a unicast message to that specific MAC address to restart authentication.

Options

interface <[*interface-name*]>—Reset the authentication state of all the supplicants (also, clear all the authentication bypassed clients) connected to the specified interface (when the interface is an authenticator) or reset the authentication state for the interface itself (when the interface is a supplicant).

mac-address [*mac-addresses*]—Reset the authentication state of the specified MAC addresses.

statistics <interface *interface-name*>—Clear 802.1X statistics on all 802.1X-enabled interfaces. If the **interface** option is specified, clear 802.1X firewall statistics for that interface or interfaces.

Required Privilege Level

view

Related Documentation

- dot1x on page 916

List of Sample Output

clear dot1x interface on page 1166
clear dot1x mac-address on page 1166
clear dot1x statistics interface on page 1167

Sample Output

clear dot1x interface

```
user@host> clear dot1x interface ge-0/0/1
```

clear dot1x mac-address

```
user@host> clear dot1x mac-address 00:04:ae:cd:23:5f
```

clear dot1x statistics interface

```
user@host> clear dot1x statistics interface ge-0/0/1
```

clear edge-virtual-bridging

Syntax

clear edge-virtual-bridging
<edge-control-protocol-statistics>
<firewall <interface *interface-name*>
<vsi-profiles <interface *interface-name*>

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description

Clear edge-virtual-bridging (EVB).

Options

none—Clear EVB.

edge-control-protocol-statistics—(Optional) Clear Edge Control Protocol (ECP) statistics.

firewall <interface *interface-name*>—(Optional) Clear EVB implicit filter counters on all interfaces or on a specific interface.

Required Privilege Level

system—To view this statement in the configuration.
system-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859
- Configuring Edge Virtual Bridging on an EX Series Switch (CLI Procedure) on page 867

clear error mac-rewrite

Syntax clear error mac-rewrite
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 9.1.

Description Clear a MAC rewrite error condition caused by the reception of tunneled protocol packets on an interface with Layer 2 protocol tunneling enabled.

On interfaces with L2PT configured, customer-facing ports should not receive packets with the L2PT MAC address as the destination address unless there is a network tolopogy or configuration error. Any such interface receiving an L2PT packet becomes “Disabled”, and must subsequently be re-enabled by clearing the error with this command.

Options interface *interface-name*—(Optional) Clear the MAC rewrite error condition for the specified interface.

Required Privilege Level clear

Related Documentation

- *Layer 2 Protocol Tunneling Through a Network*
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398
- Clearing a MAC Rewrite Error on an Interface with Layer 2 Protocol Tunneling on page 394
- show mac-rewrite interface on page 1395

List of Sample Output clear error mac-rewrite interface on page 1169

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear error mac-rewrite interface

```
user@host> clear error mac-rewrite interface ge-1/0/1
```

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 on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395

List of Sample Output clear ethernet-switching layer2-protocol-tunneling error on page 1170
clear ethernet-switching layer2-protocol-tunneling error interface xe-0/0/1.0 on page 1170

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 1224
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395

List of Sample Output
clear ethernet-switching layer2-protocol-tunneling statistics on page 1171
clear ethernet-switching layer2-protocol-tunneling error interface ge-0/1/1.0 on page 1171
clear ethernet-switching layer2-protocol-tunneling error vlan v2 on page 1171

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 recovery-timeout

Syntax clear ethernet-switching recovery-timeout
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 15.1X49-D70

Description Clear all MAC limiting errors from all the Ethernet switching interfaces on the device or from the specified interface, and restore the interfaces or the specified interface to service.

Options **interface *interface-name***—(Optional) Clear all MAC limiting errors from the specified interface and restore the interface to service.

Required Privilege Level clear

Related Documentation

- Understanding MAC Limiting on page 803
- Example: Configuring MAC Limiting on a Security Device on page 805
- Configuring Autorecovery From the Disabled State on Secure Interfaces of a Security Device (CLI Procedure) on page 808

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 on page 1243

List of Sample Output clear ethernet-switching table on page 1174

Output Fields This command produces no output.

Sample Output

clear ethernet-switching table

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

clear interfaces statistics swfabx

Syntax clear interfaces statistics <swfab0 | swfab1>

Release Information Command introduced in Junos OS Release 11.1.

Description Clear interface statistics for the specified swfab interface.

Required Privilege Level clear

Related Documentation
- show interfaces swfabx on page 1393

List of Sample Output clear interfaces statistics <swfab0 | swfab1> on page 1175

Output Fields When you enter this command, interface statistics for swfab0 and swfab1 are cleared.

Sample Output

clear interfaces statistics <swfab0 | swfab1>

```
user@host> clear interfaces statistics <swfab0 | swfab1>
```

clear mvrp statistics

List of Syntax

Syntax (EX Series) on page 1176
Syntax (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320) on page 1176

Syntax (EX Series)

clear mvrp statistics <interface *interface-name*>

Syntax (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

clear mvrp statistics
<interface interface-name>
<routing-instance routing-instance-name>

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Statement introduced in Junos OS Release 15.1X49-D70.

Description

Clear all Multiple VLAN Registration Protocol (MVRP) interface and, for SRX devices, routing instances statistics.

Options

none—Clear all MVRP statistics.

interface *interface-name*—Clear the MVRP statistics on the specified interface.

routing-instance name—Clear the MVRP statistics on the specified SRX Series device's named routing instance. If you do not specify a routing instance name, statistics are cleared for the default routing instance.

Required Privilege Level

clear

Related Documentation

- show mvrp statistics on page 1408
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- show mvrp on page 1397

List of Sample Output

clear mvrp statistics on page 1176
clear mvrp statistics interface ge-0/0/1.0 on page 1177

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear mvrp statistics

```
user@switch> clear mvrp statistics
```

clear mvrp statistics interface ge-0/0/1.0

```
user@switch> clear mvrp statistics interface ge-0/0/1.0
```

clear oam ethernet connectivity-fault-management path-database

Syntax clear oam ethernet connectivity-fault-management path-database maintenance-domain *md-name* maintenance-association *ma-name* host <*mac-addr*>

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Clear the relevant path information from the database for the specified remote host.

Options **host**—MAC address of remote host in xx:xx:xx:xx:xx:xx format.

maintenance-association —Name of the maintenance association.

maintenance-domain —Name of the maintenance domain.

Required Privilege Level clear

Related Documentation
- show oam ethernet connectivity-fault-management path-database on page 1427

List of Sample Output clear oam ethernet connectivity-fault- management path-database on page 1178

Sample Output

clear oam ethernet connectivity-fault- management path-database

```
user@host> clear oam ethernet connectivity-fault-management path-database
maintenance-domain private maintenance-association private-ma 00:00:5E:00:53:AA
Path database entries cleared for the remote-host
```

clear oam ethernet connectivity-fault-management statistics

Syntax clear oam ethernet connectivity-fault-management statistics
interface
level

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Clear connectivity fault management (CFM) statistics.

Options **Interface**—Clear the statistics on an interface.

Level—The maintenance-domain level (0 through 7).

Required Privilege Level View

Related Documentation
- show oam ethernet connectivity-fault-management mep-statistics on page 1422

List of Sample Output clear oam ethernet connectivity-fault- management statistics on page 1179

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear oam ethernet connectivity-fault- management statistics

```
user@host> clear oam ethernet connectivity-fault-management statistics
Cleared statistics of all CFM sessions
```

clear security flow ip-action

Syntax **clear security flow ip-action** [*filter*]

Release Information Command introduced in Junos OS Release 10.4. Logical systems option introduced in Junos OS Release 11.2.

Description Clear IP-action entries, based on filtered options, for IP sessions running on the device.

Options *filter*—Filter the display based on the specified criteria.

The following filters display those sessions that match the criteria specified by the filter. Refer to the sample output for filtered output examples.

all | [*filter*]—All active sessions on the device.

destination-port ***destination-port***—Destination port number of the traffic. Range is 1 through 65,535.

destination-prefix ***destination-prefix***—Destination IP prefix or address.

family (inet | inet6) [*filter*]—IPv4 traffic or IPv6-NATPT traffic and filtered options.

logical-system ***logical-system-name*** **| all** [*filter*]—Specified logical system or all logical systems.

protocol ***protocol-name*** **|** ***protocol-number*** [*filter*]—Protocol name or number and filtered options.

- **ah** or **51**
- **egp** or **8**
- **esp** or **50**
- **gre** or **47**
- **icmp** or **1**
- **icmp6** or **58**
- **ipip** or **4**
- **ospf** or **89**
- **pim** or **103**
- **rsvp** or **46**
- **sctp** or **132**
- **tcp** or **6**
- **udp** or **17**

root-logical-system [*filter*]—Default logical system information and filtered options.

source-port *source-port*—Source port number of the traffic. Range is 1 through 65,535.

source-prefix *source-prefix*—Source IP prefix or address of the traffic.

Required Privilege Level clear

Related Documentation
- show security flow ip-action on page 1470

List of Sample Output
clear security flow ip-action all on page 1181
clear security flow ip-action destination-prefix on page 1181
clear security flow ip-action family inet on page 1181
clear security flow ip-action protocol udp on page 1181

Output Fields When you enter this command, the system responds with the status of your request.

Sample Output

clear security flow ip-action all

```
user@host>clear security flow ip-action all
1008 ip-action entries cleared
```

clear security flow ip-action destination-prefix

```
user@host>clear security flow ip-action destination-prefix 192.0.2.5/24
87 ip-action entries cleared
```

clear security flow ip-action family inet

```
user@host>clear security flow ip-action family inet
2479 ip-action entries cleared
```

clear security flow ip-action protocol udp

```
user@host>clear security flow ip-action protocol udp
270 ip-action entries cleared
```

clear security flow session family

Syntax clear security flow session family (inet | inet6)

Release Information Command introduced in Junos OS Release 10.2.

Description Clear sessions that match the specified protocol family.

Options
- **inet**—Clear IPv4 sessions.
- **inet6**—Clear IPv6 sessions.

Required Privilege Level clear

Related Documentation
- show security flow session family on page 1478

List of Sample Output
clear security flow session family inet on page 1182
clear security flow session family inet6 on page 1182

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear security flow session family inet

```
user@host> clear security flow session family inet
1 active sessions cleared
```

clear security flow session family inet6

```
user@host> clear security flow session family inet6
1 active sessions cleared
```

show chassis cluster ethernet-switching interfaces

Syntax show chassis cluster ethernet-switching interfaces

Release Information Command introduced in Junos OS Release 11.1.

Description Display the status of the switch fabric interfaces (swfab interfaces) in a chassis cluster.

Required Privilege Level view

Related Documentation
- *cluster (Chassis)*
- *Ethernet Switching Feature Guide*

List of Sample Output show chassis cluster ethernet-switching interfaces on page 1183

Output Fields Table 122 on page 1183 lists the output fields for the **show chassis cluster ethernet-switching interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 122: show chassis cluster ethernet-switching interfaces Output Fields

Field Name	Field Description
swfab ***switch fabric interface-name***	Name of the switch fabric interface. • Name—Name of the physical interface. • Status—State of the switch fabric interface: up or down.

Sample Output

show chassis cluster ethernet-switching interfaces

```
user@host> show chassis cluster ethernet-switching interfaces
swfab0:
    Name                Status
    ge-0/0/9            up
    ge-0/0/10           up
swfab1:
    Name                Status
    ge-7/0/9            up
    ge-7/0/10           up
```


show chassis cluster ethernet-switching status

Syntax show chassis cluster ethernet-switching status

Release Information Command introduced in Junos OS Release 11.1.

Description Display the Ethernet switching status of the chassis cluster.

Required Privilege Level view

Related Documentation
- *cluster (Chassis)*
- *Ethernet Switching Feature Guide*

List of Sample Output show chassis cluster ethernet-switching status on page 1185

Output Fields Table 123 on page 1184 lists the output fields for the **show chassis cluster ethernet-switching status** command. Output fields are listed in the approximate order in which they appear.

Table 123: show chassis cluster ethernet-switching status Output Fields

Field Name	Field Description
Cluster ID	ID number (1-255) of a cluster. Setting a cluster ID to 0 is equivalent to disabling a cluster. A cluster ID greater than 15 can only be set when the fabric and control link interfaces are connected back-to-back.
Redundancy-Group	You can create up to 128 redundancy groups in the chassis cluster.
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
Priority	Assigned priority for the redundancy group on that node.
Status	State of the redundancy group (**Primary**, **Secondary**, **Lost**, or **Unavailable**). • **Primary**—Redundancy group is active and passing traffic. • **Secondary**—Redundancy group is passive and not passing traffic. • **Lost**—Node loses contact with the other node through the control link. Most likely to occur when both nodes are in a cluster and there is a control link failure, one node cannot exchange heartbeats, or when the other node is rebooted. • **Unavailable**—Node has not received a single heartbeat over the control link from the other node since the other node booted up. Most likely to occur when one node boots up before the other node, or if only one node is present in the cluster.
Preempt	• **Yes**: Mastership can be preempted based on priority. • **No**: Change in priority will not preempt mastership.

Table 123: show chassis cluster ethernet-switching status Output Fields (continued)

Field Name	Field Description
Manual failover	• **Yes**: Mastership is set manually through the CLI. • **No**: Mastership is not set manually through the CLI.

Sample Output

show chassis cluster ethernet-switching status

```
user@host> show chassis cluster ethernet-switching status

Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 0
node0  1      primary        no      no       None
node1  1      secondary      no      no       None

Ethernet switching status:
    Probe state is UP. Both nodes are in single ethernet switching domain(s).
```

show chassis cluster status

Syntax **show chassis cluster status**
<redundancy-group *group-number* >

Release Information Support for monitoring failures added in Junos OS Release 12.1X47-D10.

Description Display the current status of the Chassis Cluster. You can use this command to check the status of chassis cluster nodes, redundancy groups, and failover status.

Options
- none—Display the status of all redundancy groups in the chassis cluster.
- **redundancy-group *group-number***—(Optional) Display the status of the specified redundancy group.

Required Privilege Level view

Related Documentation
- *redundancy-group (Chassis Cluster)*
- *clear chassis cluster failover-count*
- *request chassis cluster failover node*
- *request chassis cluster failover reset*

List of Sample Output
show chassis cluster status on page 1187
show chassis cluster status with preemptive delay on page 1188
show chassis cluster status redundancy-group 1 on page 1188

Output Fields Table 124 on page 1186 lists the output fields for the **show chassis cluster status** command. Output fields are listed in the approximate order in which they appear.

Table 124: show chassis cluster status Output Fields

Field Name	Field Description
Cluster ID	ID number (1-15) of a cluster is applicable for releases upto Junos OS Release 12.1X45-D10. ID number (1-255) is applicable for Releases 12.1X45-D10 and later. Setting a cluster ID to 0 is equivalent to disabling a cluster.
Redundancy-Group	You can create up to 128 redundancy groups in the chassis cluster.
Node name	Node (device) in the chassis cluster (**node0** or **node1**).
Priority	Assigned priority for the redundancy group on that node.

Table 124: show chassis cluster status Output Fields (continued)

Field Name	Field Description
Status	State of the redundancy group (**Primary**, **Secondary**, **Lost**, or **Unavailable**). • **Primary**—Redundancy group is active and passing traffic. • **Secondary**—Redundancy group is passive and not passing traffic. • **Lost**—Node loses contact with the other node through the control link. Most likely to occur when both nodes are in a cluster and there is a control link failure, one node cannot exchange heartbeats, or when the other node is rebooted. • **Unavailable**—Node has not received a single heartbeat over the control link from the other node since the other node booted up. Most likely to occur when one node boots up before the other node, or if only one node is present in the cluster.
Preempt	• **Yes**: Primary state can be preempted based on priority. • **No**: Change in priority will not preempt the primary state.
Manual failover	• **Yes**: Primary state is set manually through the CLI with the **request chassis cluster failover node** or **request chassis cluster failover redundancy-group** command. This overrides **Priority** and **Preempt**. • **No**: Primary state is not set manually through the CLI.
Monitor-failures	• **None**: Cluster working properly. • **Monitor Failure code**: Cluster is not working properly and the respective failure code is displayed.

Sample Output

show chassis cluster status

```
user@host> show chassis cluster status

Monitor Failure codes:
  CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 0 , Failover count: 1
node0  200      primary        no      no       None
node1  1        secondary      no      no       None

Redundancy group: 1 , Failover count: 1
node0  101      primary        no      no       None
node1  1        secondary      no      no       None
```

Sample Output

show chassis cluster status with preemptive delay

```
user@host> show chassis cluster status

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures
Redundancy group: 0, Failover count: 1
node0  200      primary        no      no       None
node1  100      secondary      no      no       None
Redundancy group: 1, Failover count: 3
node0 200    primary-preempt-hold    yes   no    None node1  100      secondary
                yes      no       None
```

Sample Output

show chassis cluster status redundancy-group 1

```
user@host> show chassis cluster status redundancy-group 1

Monitor Failure codes:
    CS  Cold Sync monitoring        FL  Fabric Connection monitoring
    GR  GRES monitoring             HW  Hardware monitoring
    IF  Interface monitoring        IP  IP monitoring
    LB  Loopback monitoring         MB  Mbuf monitoring
    NH  Nexthop monitoring          NP  NPC monitoring
    SP  SPU monitoring              SM  Schedule monitoring
    CF  Config Sync monitoring

Cluster ID: 1
Node   Priority Status         Preempt Manual   Monitor-failures

Redundancy group: 1 , Failover count: 1
node0  101      primary        no      no       None
node1  1        secondary      no      no       None
```

show chassis forwarding-options

Syntax show chassis forwarding-options

Release Information Command introduced in Junos OS Release 13.2
Support added to QFX5200 switches in Junos OS Release 15.1X53-D30

Description Display the configuration for the Unified Forwarding Table.

Options There are no options for this command.

NOTE: Starting in Junos OS Releases 17.3R2, for QFX5200 Virtual Chassis, information about memory banks are displayed only for the Master, not for the other members. Values remain the same across all members. All configuration changes for the Unified Forwarding Table are made through the Master.

Required Privilege Level view

Related Documentation

- Configuring the Unified Forwarding Table on Switches on page 37
- Example: Configuring a Unified Forwarding Table Custom Profile on QFX Series Switches on page 47

List of Sample Output

show chassis forwarding-options (l2-profile-three) on page 1190
show chassis forwarding-options (custom-profile on QFX5200 Series switch) on page 1190
show chassis forwarding-options (QFX5200 Virtual Chassis) on page 1191

Output Fields Table 125 on page 1189 lists the output fields for the **show chassis forwarding-options** command. Output fields are listed in the approximate order in which they appear.

Table 125: show chassis forwarding-options Output Fields

Field Name	Field Description
profile name	Name of profile configured: • **custom-profile** (QFX5200 only) • **l2-profile-one** • **l2-profile-three** (default) • **l2-profile-two** • **l3-profile** • **lpm-profile**

Table 125: show chassis forwarding-options Output Fields (continued)

Field Name	Field Description
MAC	Maximum amount of memory allocated for Layer 2 entries.
L3-host	Maximum amount of memory allocated for Layer 3 host entries.
LPM	Maximum amount of memory allocated for longest match prefix (LPM) entries.
num-65-127-prefix	Maximum amount of memory allocated in LPM table for IP prefixes with lengths in the range /65 through /127.
Total scale(K)	(QFX5200 only) Maximum amount of memory allocated for each address type. This amount includes the amount configured plus the amount allocated through the dedicated hash table.
Bank details for various types of entries	(QFX5200 only) Maximum amount of memory configured by address type for each of the four shared memory banks and the dedicated hash table.
Entry type	(QFX5200 only) Type of forwarding-table entry: **L2(mac)**; **L3 (unicast and multicast)**; **Exact Match**; and **Longest Prefix Match (lpm)**
Dedicated bank size(K)	(QFX5200 only) Maximum amount of memory allocated for each address type in the dedicated hash table.
Shared bank size(K)	(QFX5200 only) Default Maximum amount of memory allocated for each address type in the shared memory banks.

Sample Output

show chassis forwarding-options (l2-profile-three)

```
user@host> show chassis forwarding-options
UFT Configuration:
l2-profile-three. (MAC: 160K L3-host: 144K LPM: 16K) (default)
num-65-127-prefix = none

{master:0}
```

show chassis forwarding-options (custom-profile on QFX5200 Series switch)

```
user@host> show chasiss forwarding-options
UFT Configuration:
custom-profile
Configured custom scale:
Entry type                 Total scale(K)
L2(mac)                         8
L3 (unicast & multicast)        72
Exact Match                     0
Longest Prefix Match (lpm)      80
num-65-127-prefix = 1K
-------------Bank details for various types of entries------------
Entry type                   Dedicated Bank Size(K)      Shared Bank Size(K)
L2 (mac)                     8                          32 * num shared banks
L3 (unicast & multicast      8                          32 * num shared banks
```

```
                        Exact match                 0                       16 * num shared banks
                        Longest Prefix match(lpm)   16                      32 * num shared banks
```

show chassis forwarding-options (QFX5200 Virtual Chassis)

```
user@host> show chassis forwarding-options

localre:
-
UFT Configuration:
l2-profile-three.(default)
num-65-127-prefix = 1K
-Bank details for various types of entries-
Entry type                          Dedicated Bank Size(K)     Shared Bank Size(K)
L2(mac)                             8                          32 * num shared banks
L3(unicast & multicast)             8                          32 * num shared banks
Exact Match                         0                          16 * num shared banks
Longest Prefix Match(lpm)           16                         32 * num shared banks

fpc1:
-
UFT Configurtion:
l2-profile-three.(default)
num-65-127-prefix = 1K
```

show dot1x authentication-bypassed-users

Syntax show dot1x authentication-bypassed-users

Release Information Command introduced in Junos OS Release 15.1X49-D80.

Description Display the supplicants (users) that have bypassed 802.1X authentication.

Required Privilege Level view

Related Documentation
- show dot1x authentication-failed-users on page 1193
- dot1x on page 916

List of Sample Output show dot1x authentication-bypassed-users on page 1192

Output Fields Table 126 on page 1192 lists the output fields for the show dot1x authentication-bypassed-users command. Output fields are listed in the approximate order in which they appear.

Table 126: show dot1x authentication-bypassed-users Output Fields

Field Name	Field Description	Level of Output
Interface	The MAC address configured to bypass 802.1X authentication.	**all**
MAC address	The MAC address configured statically on the interface.	**all**
VLAN	The VLAN that is configured to bypass 802.1X authentication.	**all**

Sample Output

show dot1x authentication-bypassed-users

```
user@host> show dot1x authentication-bypassed-users

MAC address           Interface      VLAN
00:50:56:85:66:0f     ge-0/0/0.0     vlan6
00:50:56:9e:56:42     ge-0/0/1.0     vlan6
```

show dot1x authentication-failed-users

Syntax show dot1x authentication-failed-users

Release Information Command introduced in Junos OS Release 15.1X49-D80.

Description Display the supplicants (users) that have failed 802.1X authentication.

Required Privilege Level view

Related Documentation
- show dot1x authentication-bypassed-users on page 1192
- dot1x on page 916

List of Sample Output show dot1x authentication-failed-users on page 1193

Output Fields Table 127 on page 1193 lists the output fields for the **show dot1x authentication-failed-users** command. Output fields are listed in the approximate order in which they appear.

Table 127: show dot1x authentication-failed-users Output Fields

Field Name	Field Description	Level of Output
Interface	The MAC address configured to bypass 802.1X authentication.	**all**
MAC address	The MAC address configured statically on the interface.	**all**
User	The user that is configured on the RADIUS server and that has failed 802.1X authentication.	**all**
Failure Count	The number of times that 802.1X authentication has failed on the interface.	**all**

Sample Output

show dot1x authentication-failed-users

```
user@host> show dot1x authentication-failed-users

Interface     MAC address          User            Failure Count
ge-0/0/0.0    00:50:56:85:66:0f    00505685660f    1
ge-0/0/1.0    00:50:56:9e:56:42    0050569e5642    1
```

show dot1x interface

Syntax **show dot1x interface** <<*interface-name*>
<brief | detail>

Release Information Command introduced in Junos OS Release 15.1X49-D80.

Description Display the current operational state of all ports with the list of connected users.

This command displays the list of connected supplicants received from the RADIUS authentication server regardless of the session state—that is, for both authenticated supplicants and for supplicants that attempted authentication.

Options **none**—Display information for all authenticator ports.

brief | detail—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display information for the specified interface with a list of connected supplicants.

Required Privilege Level view

Related Documentation

- show dot1x authentication-bypassed-users on page 1192
- dot1x on page 916

List of Sample Output

show dot1x interface brief on page 1198
show dot1x interface detail on page 1198

Output Fields Table 128 on page 1194 lists the output fields for the **show dot1x interface** command. Output fields are listed in the approximate order in which they appear.

Table 128: show dot1x interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a port.	**all**
MAC address	The MAC address of the connected supplicant on the port.	**all**
Role	The 802.1X authentication role of the interface. When 802.1X is enabled on an interface, the role is **Authenticator**. As **Authenticator**, the interface blocks LAN access until a supplicant is authenticated through 802.1X or MAC RADIUS authentication.	**brief**, **detail**

Table 128: show dot1x interface Output Fields (continued)

Field Name	Field Description	Level of Output
State	The state of the port: • **Authenticated**—The supplicant has been authenticated through the RADIUS server or has been permitted access through server fail fallback. • **Authenticating**—The supplicant is authenticating through the RADIUS server. • **Held**—An action has been triggered through server fail fallback during a RADIUS server timeout. A supplicant is denied access, permitted access through a specified VLAN, or maintains the authenticated state granted to it before the RADIUS server timeout occurred.	**brief**
User	The username of the connected supplicant.	**brief**
Administrative state	The administrative state of the port: • **auto**—Traffic is allowed through the port based on the authentication result (by default). • **force-authorize**—All traffic flows through the port irrespective of the authentication result. This state is not allowed on an interface whose VLAN membership has been set to **dynamic**. • **force-unauthorize**—All traffic drops on the port irrespective of the authentication result. This state is not allowed on an interface whose VLAN membership has been set to **dynamic**.	**detail**
Supplicant	The mode for the supplicant: • **single**—Only the first supplicant is authenticated. All other supplicants that connect later to the port are allowed full access without any further authentication. They effectively *piggyback* on the first supplicant's authentication. • **single-secure**—Only one supplicant is allowed to connect to the port. No other supplicant is allowed to connect until the first supplicant logs out. • **multiple**—Multiple supplicants are allowed to connect to the port. Each supplicant is authenticated individually.	**detail**
Quiet period	The number of seconds the port waits before reattempting authentication after a failed authentication exchange with the supplicant.	**detail**
Transmit period	The number of seconds the port waits before retransmitting the initial EAPOL PDUs to the supplicant.	**detail**
MAC Radius	MAC RADIUS authentication: • **enabled**—The device sends an EAPOL request to the connecting host to attempt 802.1X authentication and if the connecting host is unresponsive, the device tries to authenticate the host by using the MAC address. • **disabled**—The default. The device does not attempt to authenticate the MAC address of the connecting host.	**detail**

Table 128: show dot1x interface Output Fields (continued)

Field Name	Field Description	Level of Output
MAC Radius authentication protocol	MAC RADIUS authentication protocol: • **EAP-MD5**—The EAP-MD5 protocol is used for MAC RADIUS authentication. EAP-MD5 is an authentication method belonging to the Extensible Authentication Protocol (EAP) authentication framework. EAP-MD5 is the default authentication protocol. • **PAP**—The Password Authentication Protocol (PAP) authentication protocol is used for MAC RADIUS authentication.	**detail**
MAC Radius restrict	The authentication method is restricted to MAC RADIUS. 802.1X authentication is not enabled.	**detail**
Reauthentication	The reauthentication state: • **disable**—Periodic reauthentication of the client is disabled. • **interval**—Sets the periodic reauthentication time interval.	**detail**
Supplicant timeout	The number of seconds the port waits for a response when relaying a request from the authentication server to the supplicant before resending the request.	**detail**
Server timeout	The number of seconds the port waits for a reply when relaying a response from the supplicant to the authentication server before timing out.	**detail**
Maximum EAPOL requests	The maximum number of times an EAPOL request packet is retransmitted to the supplicant before the authentication session times out.	**detail**
Number of clients bypassed because of authentication	The number of non-802.1X clients granted access to the LAN by means of static MAC bypass. The following fields are displayed: • **Client**—MAC address of the client. • **vlan**—The name of the VLAN to which the client is connected.	**detail**
Guest VLAN member	The VLAN to which a supplicant is connected when the supplicant is authenticated using a guest VLAN. If a guest VLAN is not configured on the interface, this field displays <**not configured**>.	**detail**
Number of connected supplicants	The number of supplicants connected to a port.	**detail**
Supplicant	The username and MAC address of the connected supplicant.	**detail**

Table 128: show dot1x interface Output Fields (continued)

Field Name	Field Description	Level of Output

Authentication method	The authentication method used for a supplicant:
	• **CWA Authentication**—A supplicant is authenticated by the central Web authentication (CWA) server.
	• **Guest VLAN**—A supplicant is connected to the LAN through the guest VLAN.
	• **MAC RADIUS**—A nonresponsive host is authenticated based on its MAC address. The MAC address is configured as permitted on the RADIUS server. The RADIUS server lets the device know that the MAC address is a permitted address, and the device opens LAN access to the nonresponsive host on the interface to which it is connected.
	• **RADIUS**—A supplicant is configured on the RADIUS server, the RADIUS server communicates this to the device, and the device opens LAN access on the interface to which the supplicant is connected.
	• **Server-fail**—One of the following fallback actions is in effect because the RADIUS server is unreachable. Indicates whether EAPOL block is in effect, and the amount of time remaining for EAPOL block (in seconds).
	  • **deny**—The supplicant is denied access to the LAN, preventing traffic from flowing from the supplicant through the interface. This is the default server fail fallback action.
	  • **permit**—The supplicant is permitted access to the LAN as if the supplicant had been successfully authenticated by the RADIUS server.
	  • **use-cache**—In the event that the RADIUS server times out when the supplicant is attempting reauthentication, the supplicant is reauthenticated only if it was previously authenticated; otherwise, the supplicant is denied LAN access.
	  • **VLAN**—The supplicant is configured to be moved to a specified VLAN if the RADIUS server is unavailable to reauthenticate the supplicant. (The VLAN must already exist on the device.)
		detail

Authenticated VLAN	The VLAN to which the supplicant is connected.	**detail**
Dynamic filter	User policy filter sent by the RADIUS server.	**detail**
Session Reauth interval	The configured reauthentication interval.	**detail**
Reauthentication due in	The number of seconds in which reauthentication occurs again for the connected supplicant.	**detail**
Session Accounting Interim Interval	The number of seconds between interim RADIUS accounting messages.	**detail**
Accounting Update due in	The number of seconds until the next interim RADIUS accounting update is due.	**detail**
CWA Redirect URL	The URL used to redirect the supplicant to a central Web server for authentication.	**detail**

Sample Output

show dot1x interface brief

```
user@root> show dot1x interface brief
802.1X Information:
Interface     Role           State           MAC address          User
ge-0/0/1      Authenticator  Connecting      00:50:56:85:66:0F    00505685660f
ge-0/0/2      Authenticator  Authenticated   00:50:56:9E:56:42    0050569e5642
```

show dot1x interface detail

```
user@root> show dot1x interface detail

 ge-0/0/0.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Multiple
  Number of retries: 3
  Quiet period: 60 seconds
  Transmit period: 30 seconds
  Mac Radius: Enabled
  Mac Radius Restrict: Disabled
  Reauthentication: Enabled
  Configured Reauthentication interval: 30 seconds
  Supplicant timeout: 30 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 2
  Guest VLAN member: not configured
  Number of connected supplicants: 1
    Supplicant: 00505685660f, 00:50:56:85:66:0F
      Operational state: Authenticated
      Backend Authentication state: Idle
      Authentication method: Server-Reject Vlan
      Authenticated VLAN: visitor-vlan
      Session Reauth interval: 30 seconds
      Reauthentication due in 20 seconds
ge-0/0/1.0
  Role: Authenticator
  Administrative state: Auto
  Supplicant mode: Multiple
  Number of retries: 3
  Quiet period: 60 seconds
  Transmit period: 30 seconds
  Mac Radius: Enabled
  Mac Radius Restrict: Disabled
  Reauthentication: Enabled
  Configured Reauthentication interval: 30 seconds
  Supplicant timeout: 30 seconds
  Server timeout: 30 seconds
  Maximum EAPOL requests: 2
  Guest VLAN member: not configured
  Number of connected supplicants: 1
    Supplicant: 0050569e5642, 00:50:56:9E:56:42
      Operational state: Authenticated
      Backend Authentication state: Idle
      Authentication method: Server-Reject Vlan
      Authenticated VLAN: visitor-vlan
      Session Reauth interval: 30 seconds
      Reauthentication due in 24 seconds
```

show dot1x static-mac-address

Syntax show dot1x static-mac-address <interface *interface-name*>

Release Information Command introduced in Junos OS Release 15.1X49-D80.

Description Display all the static MAC addresses of interfaces that are configured to bypass 802.1X authentication.

Options **none**—Display static MAC addresses for all interfaces.

interface *interface-name*—(Optional) Display static MAC addresses for a specific interface.

Required Privilege Level view

Related Documentation
- show dot1x authentication-bypassed-users on page 1192
- dot1x on page 916

List of Sample Output
show dot1x static-mac-address on page 1200
show dot1x static-mac-address interface (Specific Interface) on page 1201

Output Fields Table 129 on page 1200 lists the output fields for the **show dot1x static-mac-address** command. Output fields are listed in the approximate order in which they appear.

Table 129: show dot1x static-mac-address Output Fields

Field Name	Field Description	Level of Output
MAC address/prefix	The MAC address of the device that is configured to bypass 802.1X authentication.	**all**
VLAN-Assignment	The name of the VLAN to which the device is assigned.	**all**
Interface	The name of the interface on which authentication is bypassed for a given MAC address.	**all**

Sample Output

show dot1x static-mac-address

```
user@host> show dot1x static-mac-address

MAC address/prefix          VLAN-Assignment          Interface
00:50:56:85:66:0f/48        vlan6                    ge-0/0/0.0
00:50:56:9e:56:42/48        vlan6                    ge-0/0/1.0
```

show dot1x static-mac-address interface (Specific Interface)

```
user@host> show dot1x static-mac-address interface ge-0/0/0

MAC address/prefix          VLAN-Assignment          Interface
00:50:56:85:66:0f/48        vlan6                    ge-0/0/0.0
```

show dot1x statistics

Syntax **show dot1x statistics interface <*interface-name*>**

Release Information Command introduced in Junos OS Release 15.1X49-D80.

Description Display 802.1X statistics on this interface.

Options **interface *interface-name***—(Optional) Displays statistical information for the interface.

Required Privilege Level view

Related Documentation
- dot1x on page 916
- show dot1x authentication-bypassed-users on page 1192

List of Sample Output show dot1x statistics interface on page 1202

Sample Output

show dot1x statistics interface

```
user@host> show dot1x statistics interface ge-0/0/0

Interface: ge-0/0/0.0
  TxReqId = 4 TxReq = 0 TxTotal = 4
  RxStart = 0 RxLogoff = 0 RxRespId = 0 RxResp = 0
  RxInvalid = 0 RxLenErr = 0 RxTotal = 0
  LastRxVersion = 0 LastRxSrcMac = 00:50:56:85:66:0f
```

show edge-virtual-bridging

Syntax

show edge-virtual-bridging
<detail>
<edge-control-protocol statistics <interface *interface-name*>>
<firewall>
<interface *interface-name*>
vsi-profiles <interface *interface-name*>

Release Information

Command introduced in Junos OS Release 12.1 for EX Series switches.

Description

Display information about edge virtual bridging (EVB).

Options

none—Display EVB parameters for all interfaces configured with EVB.

detail—(Optional) Display EVB parameters and virtual station interface (VSI) profiles associated with each interface.

edge-control-protocol statistics <interface *<nterface-name>*—(Optional) Display Edge Control Protocol (ECP)l statistics for all configured EVB interfaces or for the specified interface.

firewall—Display the firewall filters created by EVB.

interface *interface-name*—(Optional) Display EVB parameters for the specified interface.

vsi-profiles <interface *interface-name*>—(Optional) Display VSI profiles associated on each interface or for the specified interface.

Required Privilege Level

view

Related Documentation

- Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859Example: Configuring Edge Virtual Bridging for Use with VEPA Technology on an EX Series Switch on page 859

List of Sample Output

Output Fields

Table 130 on page 1204 lists the output fields for the **show edge-virtual-bridging** command. Output fields are listed in the approximate order in which they appear.

Table 130: show edge-virtual-bridging Output Field Descriptions

Field Name	Field Description
Interface	Switch interface configured for EVB.
Interface input ECP Packets	Number of ECP packets received by the switch. ECP is a Layer 2 protocol that is used to carry VSI Discovery and Configuration Protocol (VDP) messages.
Interface output ECP Packets	Number of ECP packets sent by the switch. ECP is a Layer 2 protocol that is used to carry VDP messages.
Forwarding Mode	Mode by which packets are forwarded to their destination. The value for forwarding mode is either **Standard** (meaning the forwarding is done through 802.1Q) or **Reflective-relay**, meaning that both the source and destination addresses are located on the same VM server.
RTE	Retransmission timer exponent (RTE) is an EVB interface attribute used to calculate the minimum VDP protocol data unit (PDU) retransmission time.
Number of VSIs	Number of virtual station interfaces on the switch connected to the VEPA.
Protocols	EVB protocols currently enabled. The values can be **VDP**, **ECP** or **RTE**. Protocols are configured during the capabilities exchange via an EVB type, length, and value (TLV) carried by the Link Layer Discovery Protocol (LLDP) between the switch and the server.
VSI profile	EVB profile including parameters that uniquely identify each VSI entry (VSI manager, VSI type, VSI version, VSI instance, VSI state).
Filter Name	Name of the filter defined in the firewall stanza.
Counters	Number of packets and bytes that have satisfied the match conditions defined by the filter.

Sample Output

show edge-virtual-bridging

```
user@switch#show edge-virtual-bridging
Interface     Forwarding Mode      RTE   Number of VSIs  Protocols
ge-0/0/20.0   Reflective-relay     25    400             ECP, VDP, RTE
```

show edge-virtual-bridging interface

```
user@switch#show edge-virtual-bridging interface ge-0/0/20.0
Interface: ge-0/0/20.0, Forwarding mode: Reflective-relay RTE: 25, Number of VSIs:
 400, Protocols: ECP, VDP, RTE
VSI profiles:
 Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
      MAC                       VLAN
      00:10:94:00:00:04
```

show edge-virtual-bridging edge-control-protocol statistics

```
user@switch#show edge-virtual-bridging edge-control-protocol-statistics
Interface: ge-0/0/20.0
      Input ECP packets: 302
      Output ECP packets: 303
```

show edge-virtual-bridging vsi-profiles

```
user@switch#show edge-virtual-bridging vsi-profiles
Interface: ge-0/0/20.0
Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
      MAC                       VLAN
      00:10:94:00:00:04         3
```

show edge-virtual-bridging vsi-profiles interface

```
user@switch#show edge-virtual-bridging vsi-profiles interface ge-0/0/20.0
Interface: ge-0/0/20.0
Manager: 97, Type: 997, Version: 3, VSI State: Associate
 Instance: 09b11c53-8b5c-4eeb-8f00-c84ebb0bb997
          MAC
          VLAN    00:10:94:00:00:04         3
```

show edge-virtual-bridging firewall

```
user@switch#show edge-virtual-bridging firewall
Filter name: evb_filter_ge-0/0/20
Counters:
  Name: evb_filter_term_3_00:10:94:00:00:04_default
          Bytes: 0, Packets: 0
  Name: f3_accept__evb_filter_term_3_00:10:94:00:00:04-f3-t1
          Bytes: 1028, Packets: 14
```

show ethernet-switching flood

Syntax **show ethernet-switching flood**
<brief | detail | extensive>
<event-queue>
<instance *instance-name*>
<logical-system *logical-system-name*>
<route (all-ce-flood | all ve-flood | alt-root-flood | bd-flood | mlp-flood | re-flood)>
<vlan-name *vlan-name*>

Release Information Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Command introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description (EX Series switches and QFX Series switches only) Display Ethernet-switching flooding information.

Options **none**—Display all Ethernet-switching flooding information for all VLANs.

brief | detail | extensive—(Optional) Display the specified level of output.

event-queue—(Optional) Display the queue of pending Ethernet-switching flood events.

instance *instance-name*—(Optional) Display Ethernet-switching flooding information for the specified routing instance.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching flooding information for the specified logical system.

route (all-ce-flood | all ve-flood | alt-root-flood | bd-flood | mlp-flood | re-flood)—(Optional) Display the following:

- **all-ce-flood**—Display the route for flooding traffic to all customer edge routers or switches if **no-local-switching** is enabled.
- **all-ve-flood**—Display the route for flooding traffic to all VPLS edge routers or switches if **no-local-switching** is enabled.
- **alt-root-flood**—Display the Spanning Tree Protocol (STP) alt-root flooding route used for the interface.
- **bd-flood**—Display the route for flooding traffic of a VLAN if **no-local-switching** is not enabled.
- **mlp-flood**—Display the route for flooding traffic to MAC learning chips.
- **re-flood**—Display the route for Routing Engine flooding to all interfaces.

vlan-name *vlan-name*—(Optional) Display Ethernet-switching flooding information for the specified VLAN.

Required Privilege Level view

List of Sample Output

Sample Output

show ethernet-switching flood

```
user@host> show ethernet-switching flood
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 9
VEs: 0
VLAN Name: VLAN101
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057b/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12866
  0x30004/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12863
VLAN Name: VLAN102
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057c/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12875
  0x30005/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12872
VLAN Name: VLAN103
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057d/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12884
  0x30006/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12881
```

show ethernet-switching flood brief

```
user@host> show ethernet-switching flood brief
Name               Active CEs         Active VEs
__juniper_private1__ 0                0
default-switch     9                  0
```

show ethernet-switching flood detail

```
user@host> show ethernet-switching flood detail
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 9
VEs: 0
VLAN Name: VLAN101
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x3057b/51 FLOOD_GRP_COMP_NH __all_ces__     comp            12866
  0x30004/51 FLOOD_GRP_COMP_NH __re_flood__    comp            12863
```

```
VLAN Name: VLAN102
Flood Routes:
  Prefix    Type          Owner                 NhType          NhIndex
  0x3057c/51 FLOOD_GRP_COMP_NH __all_ces__      comp            12875
  0x30005/51 FLOOD_GRP_COMP_NH __re_flood__     comp            12872
VLAN Name: VLAN103
Flood Routes:
  Prefix    Type          Owner                 NhType          NhIndex
  0x3057d/51 FLOOD_GRP_COMP_NH __all_ces__      comp            12884
  0x30006/51 FLOOD_GRP_COMP_NH __re_flood__     comp            12881
```

show ethernet-switching flood extensive

```
user@host> show ethernet-switching flood extensive
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 9
VEs: 0
VLAN Name: VLAN101
  Flood route prefix: 0x3057b/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 12866
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12860
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605

  Flood route prefix: 0x30004/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 12863
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12860
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605
VLAN Name: VLAN102

  Flood route prefix: 0x3057c/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 12875
    Flooding to:
    Name             Type          NhType          Index
```

```
    __all_ces__      Group          comp            12869
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605

  Flood route prefix: 0x30005/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 12872
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12869
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605
VLAN Name: VLAN103

  Flood route prefix: 0x3057d/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 12884
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12878
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605

  Flood route prefix: 0x30006/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 12881
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            12878
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae20.0           CE             ucst            7605
VLAN Name: VLAN104
```

show ethernet-switching flood extensive (Junos Fusion Data Center with EVPN)

```
user@host> show ethernet-switching flood extensive
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 3
```

```
VEs: 3
VLAN Name: v100
  Flood route prefix: 0x3001b/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __ves__
  Flood group name: __ves__
  Flood group index: 0
  Nexthop type: comp
  Nexthop index: 1946
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            1945
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae0.0            CE             ucst            1886

  Flood route prefix: 0x3000f/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __all_ces__
  Flood group name: __all_ces__
  Flood group index: 1
  Nexthop type: comp
  Nexthop index: 1905
    Flooding to:
    Name             Type          NhType          Index
    __ves__          Group          comp            1971
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index
        vtep.32769       CORE_FACING    venh            1917
        vtep.32770       CORE_FACING    venh            1918
        vtep.32771       CORE_FACING    venh            1923
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            1945
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae0.0            CE             ucst            1886

  Flood route prefix: 0x30001/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp

        Name             Type          NhType          Index
        vtep.32769       CORE_FACING    venh            1917
        vtep.32770       CORE_FACING    venh            1918
        vtep.32771       CORE_FACING    venh            1923
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            1907
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ae12.0           CE             ucst            1681

  Flood route prefix: 0x30006/51
```

```
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: comp
  Nexthop index: 1891
    Flooding to:
    Name              Type           NhType          Index
    __ves__           Group           comp            1961
        Composition: flood-to-all
        Flooding to:
        Name              Type           NhType          Index
        vtep.32769        CORE_FACING     venh            1917
        vtep.32770        CORE_FACING     venh            1918
        vtep.32771        CORE_FACING     venh            1923
    Flooding to:
    Name              Type           NhType          Index
    __all_ces__       Group           comp            1907
        Composition: split-horizon
        Flooding to:
        Name              Type           NhType          Index
        ae12.0            CE              ucst            1681

...
```

show ethernet-switching interface

Syntax

show ethernet-switching interface
<brief | detail | extensive>
<interface-name>

Release Information

Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Command introduced in Junos OS Release 13.2x51 for QFX Series switches.

Description

Display Layer 2 learning information for all the interfaces.

Options

none—Display Ethernet-switching information for all interfaces.

brief | detail | extensive—(Optional) Display the specified level of output.

interface-name—(Optional) Display Ethernet-switching information for the specified interface.

Required Privilege Level

view

Related Documentation

List of Sample Output

show ethernet switching interface (Specific Interface) on page 1213
show ethernet-switching interface detail on page 1214

Output Fields

Table 131 on page 1212 describes the output fields for the **show ethernet-switching interface** command. Output fields are listed in the approximate order in which they appear.

Table 131: show ethernet-switching interface Output Fields

Field Name	Field Description
Logical interface	Name of the logical interface.
VLAN members	VLANs associated with this interface.
Tag	VLAN ID.
MAC limit	Number of MAC addresses that can be associated with the interface.
STP state	Spanning Tree protocol (STP) state.

Table 131: show ethernet-switching interface Output Fields (continued)

Field Name	Field Description
Logical interface flags	Status of Layer 2 learning properties for each interface: • **DL**—MAC learning is disabled. • **LH**—MAC interface limit has been reached. • **AD**—Packets are dropped after the MAC interface limit is reached. • **DN**—The MAC interface is down. • **MMAS**—The MAC interface is disabled after a MAC address move. • **SCTL**—The MAC interface is disabled after a configured storm-control level is exceeded.
Tagging	Tagging state of the VLAN.

Sample Output

show ethernet switching interface (Specific Interface)

```
user@host> show ethernet-switching inerface ae10.0
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
ae10.0                       8192                                  tagged
             VLAN70..   701   1024     Forwarding

             VLAN70..   702   1024     Forwarding

             VLAN70..   703   1024     Forwarding

             VLAN70..   704   1024     Forwarding

             VLAN70..   705   1024     Forwarding

             VLAN70..   706   1024     Forwarding

             VLAN70..   707   1024     Forwarding

             VLAN70..   708   1024     Forwarding

             VLAN70..   709   1024     Forwarding

             VLAN71..   710   1024     Forwarding

             VLAN71..   711   1024     Forwarding

             VLAN71..   712   1024     Forwarding

             VLAN71..   713   1024     Forwarding

             VLAN71..   714   1024     Forwarding

             VLAN71..   715
[...output truncated...]
```

show ethernet-switching interface detail

```
user@host> show ethernet-switching interface detail
Information for interface family:
Name: ge-1/0/3.0
  Type: IFF                              Handle: 0x8bba280
  Index: 331                             Generation: 159
                                         Flags: UP,
  IFD index: 141                         Routing/Vlan index: 4
  IFL index: 331                         Address family: 50
  Sequence number: 0                     MAC sequence number: 0
  MAC limit: 65535                       MACs learned: 0
  Static MACs learned: 0                 Non configured static MACs learned: 0
Name: ge-1/0/3.0
  Type: IFBD (static)                    Handle: 0x8bb6e00
  Index:                                 Generation: 129
  Trunk id: 0                            Flags: UP,
  IFD index:                             Routing/Vlan index: 2
  IFL index:                             Address family:
  Sequence number: 1                     MAC sequence number: 1
  MAC limit: 65535                       MACs learned: 0
  Static MACs learned: 0                 Non configured static MACs learned: 0
  VSTP index: 11                         Rewrite op:
Name: ge-1/0/3.0
  Type: IFBD (static)                    Handle: 0x8bb6f00
  Index:                                 Generation: 130
  Trunk id: 0                            Flags: UP,
  IFD index:                             Routing/Vlan index: 3
  IFL index:                             Address family:
  Sequence number: 1                     MAC sequence number: 1
  MAC limit: 65535                       MACs learned: 0
  Static MACs learned: 0                 Non configured static MACs learned: 0
  VSTP index: 11                         Rewrite op:
```

show ethernet-switching interfaces

Syntax

show ethernet-switching interfaces
<brief | detail | summary>
<interface *interface-name*>

Release Information

Command introduced in Junos OS Release 9.0 for EX Series switches.
In Junos OS Release 9.6 for EX Series switches, the following updates were made:

- **Blocking** field output was updated.
- The default view was updated to include information about 802.1Q tags.
- The **detail** view was updated to include information on VLAN mapping.

Command introduced in Junos OS Release 11.1 for the QFX Series.
In Junos OS Release 11.1 for EX Series switches, the **detail** view was updated to include reflective relay information.
Command introduced in Junos OS Release 14.1X53-D20 for OCX Series switches.

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 871Understanding Bridging and VLANs on Switches on page 84
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Example: Setting Up Bridging with Multiple VLANs on page 141
- *Understanding FCoE*
- *Interfaces Overview for Switches*
- **show ethernet-switching mac-learning-log on page 1229**
- **show ethernet-switching table on page 1243**
- *Configuring Autorecovery From the Disabled State on Secure or Storm Control Interfaces (CLI Procedure)*

List of Sample Output

show ethernet-switching interfaces on page 1218
show ethernet-switching interfaces summary on page 1219

show ethernet-switching interfaces brief on page 1219
show ethernet-switching interfaces detail on page 1219
show ethernet-switching interfaces interface-name on page 1220
show ethernet-switching interfaces on page 1220
show ethernet-switching interfaces ge-0/0/15 brief on page 1220
show ethernet-switching interfaces ge-0/0/2 detail (Blocked by RTG rtggroup) on page 1221
show ethernet-switching interfaces ge-0/0/15 detail (Blocked by STP) on page 1221
show ethernet-switching interfaces ge-0/0/17 detail (Disabled by bpdu-control) on page 1221
show ethernet-switching interfaces detail (C-VLAN to S-VLAN Mapping) on page 1221
show ethernet-switching interfaces detail (Reflective Relay Is Configured) on page 1221

Output Fields For QFX Series, QFabric, NFX Series, EX4600 and OCX1100:

Table 132 on page 1216 lists the output fields for the **show ethernet-switching interfaces** command on QFX Series, QFabric, NFX Series, EX4600 and OCX1100. Output fields are listed in the approximate order in which they appear.

Table 132: 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**
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**

Output fields for EX Series:

Table 133 on page 1217 lists the output fields for the **show ethernet-switching interfaces** command on EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 133: show ethernet-switching interfaces Output Fields

Field Name	Field Description	Level of Output
Interface	Name of a switching interface.	none, **brief**, **detail**, **summary**
Index	VLAN index internal to Junos OS.	**detail**
State	Interface state. Values are **up** and **down**.	none, **brief**, **detail**
Port mode	The **access** mode is the port mode default and works with a single VLAN. Port mode can also be **trunk**, which accepts tagged packets from multiple VLANs on other switches. The third port mode value is **tagged-access**, which accepts tagged packets from access devices.	**detail**
Reflective Relay Status	Reflective relay allows packets to use the same interface for both upstream and downstream traffic. When reflective relay has been configured, the status displayed is always **enabled** . When reflective relay is not configured, this entry does not appear in the command output.	**detail**
Ether type for the interface	Ether type is a two-octet field in an Ethernet frame used to indicate which protocol is encapsulated in the payload of an incoming Ethernet packet. Both 802.1Q packets and Q-in-Q packets use this field. The output displayed for this particular field indicates the interface's Ether type, which is used to match the Ether type of incoming 802.1Q packets and Q-in-Q packets. The indicated Ether type field is also added to the interface's outgoing 802.1Q and Q-in-Q packets.	**detail**
VLAN membership	Names of VLANs that belong to this interface.	none, **brief**, **detail**,
Tag	Number of the 802.1Q tag.	none, **brief**, **detail**,
Tagging	Specifies whether the interface forwards 802.1Q **tagged** or **untagged** traffic.	none, **brief**, **detail**,

Table 133: show ethernet-switching interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
Blocking	The forwarding state of the interface: • **unblocked**—Traffic is forwarded on the interface. • **blocked**—Traffic is not being forwarded on the interface. • **Disabled by bpdu control**—The interface is disabled due to receiving BPDUs on a protected interface. If the **disable-timeout** statement has been included in the BPDU configuration, the interface automatically returns to service after the timer expires. • **blocked by RTG**—The specified redundant trunk group is disabled. • **blocked by STP**—The interface is disabled due to a spanning-tree protocol error. • **MAC limit exceeded**—The interface is temporarily disabled due to a MAC limit error. The disabled interface is automatically restored to service when the disable timeout expires. • **MAC move limit exceeded**—The interface is temporarily disabled due to a MAC move limit error. The disabled interface is automatically restored to service when the disable timeout expires. • **Storm control in effect**—The interface is temporarily disabled due to a storm control error. The disabled interface is automatically restored to service when the disable timeout expires.	none, **brief**, **detail**,
Number of MACs learned on IFL	Number of MAC addresses learned by this interface.	**detail**
mapping	When mapping is configured, the status is one of the following C-VLAN to S-VLAN mapping types: • **dot1q-tunneled**—The interface maps all traffic to the S-VLAN (all-in-one bundling). • **native**—The interface maps untagged and priority tagged packets to the S-VLAN. • **push**—The interface maps packets to a firewall filter to an S-VLAN. • **policy-mapped**—The interface maps packets to a specifically defined S-VLAN. • *integer*—The interface maps packets to the specified S-VLAN. When mapping is not configured, this entry does not appear in the command output.	**detail**

Sample Output for QFX Series Switches, QFabric, NFX Series, EX4600 and OCX1100

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
```

Sample Output for EX Series Switches

show ethernet-switching interfaces

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

Interface    State  VLAN members        Tag   Tagging  Blocking

ae0.0        up      default                   untagged unblocked
ge-0/0/2.0   up      vlan300           300    untagged blocked by RTG (rtggroup)
ge-0/0/3.0   up      default                            blocked by STP
ge-0/0/4.0   down    default                            MAC limit exceeded
ge-0/0/5.0   down    default                            MAC move limit exceeded
ge-0/0/6.0   down    default                            Storm control in effect
ge-0/0/7.0   down    default                            unblocked
ge-0/0/13.0  up      default                   untagged unblocked
ge-0/0/14.0  up      vlan100             100   tagged   unblocked
                     vlan200             200   tagged   unblocked
ge-0/0/15.0  up      vlan100             100   tagged   blocked by STP
                     vlan200             200   tagged   blocked by STP
ge-0/0/16.0  down    default                   untagged unblocked
ge-0/0/17.0  down    vlan100             100   tagged   Disabled by bpdu-control

                     vlan200             200   tagged   Disabled by bpdu-control
```

show ethernet-switching interfaces ge-0/0/15 brief

```
user@switch> show ethernet-switching interfaces ge-0/0/15 brief
Interface   State  VLAN members   Tag   Tagging        Blocking

ge-0/0/15.0  up     vlan100        100   tagged         blocked by STP
                    vlan200        200   tagged         blocked by STP
```

show ethernet-switching interfaces ge-0/0/2 detail (Blocked by RTG rtggroup)

```
user@switch> show ethernet-switching interfaces ge-0/0/2 detail

Interface: ge-0/0/2.0, Index: 65, State: up, Port mode: Access
Ether type for the interface: 0X8100
VLAN membership:
    vlan300, 802.1Q Tag: 300, untagged, msti-id: 0, blocked by RTG(rtggroup)
Number of MACs learned on IFL: 0
```

show ethernet-switching interfaces ge-0/0/15 detail (Blocked by STP)

```
user@switch> show ethernet-switching interfaces ge-0/0/15 detail

Interface: ge-0/0/15.0, Index: 70, State: up, Port mode: Trunk
Ether type for the interface: 0X8100
VLAN membership:
    vlan100, 802.1Q Tag: 100, tagged, msti-id: 0, blocked by STP
    vlan200, 802.1Q Tag: 200, tagged, msti-id: 0, blocked by STP

Number of MACs learned on IFL: 0
```

show ethernet-switching interfaces ge-0/0/17 detail (Disabled by bpdu-control)

```
user@switch> show ethernet-switching interfaces ge-0/0/17 detail

Interface: ge-0/0/17.0, Index: 71, State: down, Port mode: Trunk
Ether type for the interface: 0X8100
VLAN membership:
    vlan100, 802.1Q Tag: 100, tagged, msti-id: 1, Disabled by bpdu-control
    vlan200, 802.1Q Tag: 200, tagged, msti-id: 2, Disabled by bpdu-control
Number of MACs learned on IFL: 0
```

show ethernet-switching interfaces detail (C-VLAN to S-VLAN Mapping)

```
user@switch>show ethernet-switching interfaces ge-0/0/6.0 detail
Interface: ge-0/0/6.0, Index: 73, State: up, Port mode: Access
Ether type for the interface: 0X8100
VLAN membership:
    map, 802.1Q Tag: 134, Mapped Tag: native, push, dot1q-tunneled, unblocked
    map, 802.1Q Tag: 134, Mapped Tag: 20, push, dot1q-tunneled, unblocked
```

show ethernet-switching interfaces detail (Reflective Relay Is Configured)

```
user@switch1> show ethernet-switching interfaces ge-7/0/2 detail
Interface: ge-7/0/2, Index: 66, State: down, Port mode: Tagged-access
Ether type for the interface: 0X8100
Reflective Relay Status: Enabled
Ether type for the interface: 0x8100
VLAN membership:
    VLAN_Purple VLAN_Orange VLAN_Blue, 802.1Q Tag: 450, tagged, unblocked
Number of MACs learned on IFL: 0
```

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 1224
- show ethernet-switching layer2-protocol-tunneling vlan on page 1227
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395
- show ethernet-switching layer2-protocol-tunneling statistics on page 1224
- show ethernet-switching layer2-protocol-tunneling vlan on page 1227

List of Sample Output show ethernet-switching layer2-protocol-tunneling interface on page 1223
show ethernet-switching layer2-protocol-tunneling interface xe-0/0/0.0 on page 1223

Output Fields Table 134 on page 1222 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 134: 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 1171
- show ethernet-switching layer2-protocol-tunneling interface on page 1222
- show ethernet-switching layer2-protocol-tunneling vlan on page 1227
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395
- show vlans on page 1510

List of Sample Output

show ethernet-switching layer2-protocol-tunneling statistics on page 1225
show ethernet-switching layer2-protocol-tunneling statistics interface xe-0/0/0.0 on page 1225
show ethernet-switching layer2-protocol-tunneling statistics vlan v2 on page 1225

Output Fields

Table 135 on page 1225 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 135: 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 1222
- show ethernet-switching layer2-protocol-tunneling statistics on page 1224
- Example: Configuring Layer 2 Protocol Tunneling on EX Series Switches on page 400
- Configuring Layer 2 Protocol Tunneling on EX Series Switches (CLI Procedure) on page 395
- show vlans on page 1510

List of Sample Output show ethernet-switching layer2-protocol-tunneling vlan on page 1228
show ethernet-switching layer2-protocol-tunneling vlan v2 on page 1228

Output Fields Table 136 on page 1227 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 136: 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 9.0 for EX Series switches.
Command introduced in Junos OS Release 9.5 for SRX Series devices.
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 1243
- show ethernet-switching interfaces on page 1215
- show ethernet-switching table on page 1243
- show ethernet-switching interfaces on page 1215
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Example: Connecting an EX Series Access Switch to a Distribution Switch on page 182

List of Sample Output

show ethernet-switching mac-learning-log (EX Series switch) on page 1231
show ethernet-switching mac-learning-log (QFX Series Switches, QFabric, NFX Series Devices and EX4600) on page 1231
show ethernet-switching mac-learning-log (SRX Series devices) on page 1232

Output Fields Output fields for EX Series switches:

The following table 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 137: 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.
vlan_name	VLAN name. A value defined by the user for all user-configured VLANs.
MAC	Learned MAC address.
Deleted | Added	MAC address deleted or added to the MAC learning log.

Table 137: show ethernet-switching mac-learning-log Output Fields (continued)

Field Name	Field Description
Blocking	The forwarding state of the interface: • **blocked**—Traffic is not being forwarded on the interface. • **unblocked**—Traffic is forwarded on the interface.
Flags	Displays the MAC address flags in which the MAC event occurred. This option is for debugging purposes.

Output fields for QFX Series switches, QFabric, NFX Series devices and EX4600:

Table 138 on page 1230 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 138: show ethernet-switching mac-learning-log Output Fields

Field Name	Field Description
Date and Time	Timestamp in UTC when the MAC operation occurred.
vlan_name	VLAN name. A value defined by the user for all user-configured VLANs. The name of the VLAN on which the MAC is learned.
MAC	Learned MAC address.
Event op	MAC address that are added, learned, deleted, changed or moved from one interface to another interface.
Interface Name	The name of the interface on which the MAC address is learned. When a MAC address is moved, there is another field with the name of the interface. The log displays the name of the interface from where the MAC address moved, and the name of the interface to where the MAC address moved.
Flags	Displays the MAC address flags in which the MAC event occurred. This option is for debugging purposes.

Output fields for SRX Series devices:

Table 139 on page 1230 lists the output fields for the show ethernet-switching mac-learning-log command on SRX Series devices. Output fields are listed in the approximate order in which they appear.

Table 139: 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.
VLAN-IDX	VLAN index. An internal value assigned by Junos OS for each VLAN.
MAC	Learned MAC address.

Table 139: show ethernet-switching-mac-learning-log Output Fields (continued)

Field Name	Field Description
Deleted | Added	MAC address deleted or added to the MAC learning log.
Blocking	The forwarding state of the interface: • blocked—Traffic is not being forwarded on the interface. • unblocked—Traffic is forwarded on the interface.

Sample Output

show ethernet-switching mac-learning-log (EX Series switch)

```
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-learning-log (QFX Series Switches, QFabric, NFX Series Devices and EX4600)

```
user@switch> show ethernet-switching mac-learning-log
Mon Jun 30 13:49:49 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was learned on
ge-1/0/22.0 with flags: 0x2001f     << MAC address that as dynamically learned
Mon Jun 30 13:50:29 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was deleted from
```

```
 ge-1/0/22.0 with flags: 0x1080  << MAC address that was deleted
Mon Jun 30 13:51:28 2014 vlan_name v11+11 mac 00:00:00:01:01:01 was added to
ge-1/0/22.0 with flags: 0x2013f        << Static MAC address that was added
Mon Jun 30 13:51:46 2014 vlan_name v11+11 mac 00:00:00:01:01:01 was deleted from
 ge-1/0/22.0 with flags: 0x1120   << delete of Static MAC address that was deleted
Mon Jun 30 13:52:03 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was learned on
ge-1/0/22.0 with flags: 0x2001f    << MAC address that was dynamically learned
Mon Jun 30 13:52:11 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was moved from
ge-1/0/22.0 to ge-1/0/21.0 with flags: 0x2101f    << MAC address that was moved
Mon Jun 30 13:54:24 2014 vlan_name v11+11 mac 00:10:94:00:00:02 was changed on
ge-1/0/21.0 with flags: 0x2113f   << MAC address that changed from a dynamic
address to a static address
```

show ethernet-switching mac-learning-log (SRX Series devices)

```
user@host> show ethernet-switching mac-learning-log
Wed Mar 18 08:07:05 2009
vlan_idx 7 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 9 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 10 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 11 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 12 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 13 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 14 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 15 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 16 mac 00:00:5E:00:53:00 was deleted
Wed Mar 18 08:07:05 2009
vlan_idx 4 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 6 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 7 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 9 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 10 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 11 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 12 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 13 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 14 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 15 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 16 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 5 mac 00:00:5E:00:53:00 was added
Wed Mar 18 08:07:05 2009
vlan_idx 18 mac 00:00:5E:00:53:AA was learned
Wed Mar 18 08:07:05 2009
```

```
vlan_idx 5 mac 00:00:5E:00:53:AB was learned
Wed Mar 18 08:07:05 2009
vlan_idx 6 mac 00:00:5E:00:53:AC was learned
Wed Mar 18 08:07:05 2009
vlan_idx 16 mac 00:00:5E:00:53:AD was learned
Wed Mar 18 08:07:05 2009
vlan_idx 7 mac 00:00:5E:00:53:AE was learned
Wed Mar 18 08:07:05 2009
vlan_idx 8 mac 00:00:5E:00:53:AF was learned
Wed Mar 18 08:07:05 2009
vlan_idx 12 mac 00:00:5E:00:53:AG was learned
[output truncated]
```

show ethernet-switching statistics

Syntax **show ethernet-switching statistics**
<instance *instance-name*>
<logical-system *logical-system-name*>
<vlan-name *vlan-name*>

Release Information Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.

Description (MX Series routers, QFX Series switches, and EX Series switches only) Display Ethernet-switching statistics.

Options **none**—Display Ethernet-switching statistics for all VLANs in all routing instances.

instance *instance-name*—(Optional) Display statistics for the specified routing instance.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching statistics information for the specified logical system.

vlan-name *vlan-name*—(Optional) Display statistics for the specified VLAN.

Required Privilege Level view

List of Sample Output show ethernet-switching statistics on page 1234

Sample Output

show ethernet-switching statistics

```
user@host> show ethernet-switching statistics
 Local interface: ae1.0, Index: 1035
   Broadcast packets:                   220
   Broadcast bytes  :                 13720
   Multicast packets:                   130
   Multicast bytes  :                 11700
   Flooded packets  :                     0
   Flooded bytes    :                     0
   Unicast packets  :                     0
   Unicast bytes    :                     0
   Current MAC count:                     0 (Limit 1024)
 Local interface: vt-3/3/10.1048576, Index: 1280
   Broadcast packets:                     0
   Broadcast bytes  :                     0
   Multicast packets:                     0
   Multicast bytes  :                     0
   Flooded packets  :                     2
   Flooded bytes    :                   128
   Unicast packets  :                   632
   Unicast bytes    :                 39184
   Current MAC count:                     2
 Local interface: ge-3/1/2.0, Index: 1258
   Broadcast packets:                   100
```

```
  Broadcast bytes  :                  6800
  Multicast packets:                   200
  Multicast bytes  :                 18000
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                   632
  Unicast bytes    :                 39184
  Current MAC count:                     2 (Limit 1024)
Local interface: ae3.0, Index: 1043
  Broadcast packets:                     0
  Broadcast bytes  :                     0
  Multicast packets:                     0
  Multicast bytes  :                     0
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                     0
  Unicast bytes    :                     0
  Current MAC count:                     0 (Limit 1024)
Local interface: ge-3/3/8.0, Index: 1276
  Broadcast packets:                     0
  Broadcast bytes  :                     0
  Multicast packets:                     0
  Multicast bytes  :                     0
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                     0
  Unicast bytes    :                     0
  Current MAC count:                     0 (Limit 8192)
Local interface: ae5.0, Index: 1045
  Broadcast packets:                     0
  Broadcast bytes  :                     0
  Multicast packets:                     0
  Multicast bytes  :                     0
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                     0
  Unicast bytes    :                     0
  Current MAC count:                     0 (Limit 8192)
Local interface: ae4.0, Index: 1044
  Broadcast packets:                   200
  Broadcast bytes  :                 13600
  Multicast packets:                     0
  Multicast bytes  :                     0
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                     0
  Unicast bytes    :                     0
  Current MAC count:                     0 (Limit 8192)
Local interface: ae26.0, Index: 1042
  Broadcast packets:                     0
  Broadcast bytes  :                     0
  Multicast packets:                     0
  Multicast bytes  :                     0
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                     0
  Unicast bytes    :                     0
  Current MAC count:                     0 (Limit 8192)
Local interface: ae25.0, Index: 1041
  Broadcast packets:                   133
  Broadcast bytes  :                  7980
```

```
  Multicast packets:                369934
  Multicast bytes  :              59207572
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                  1433
  Unicast bytes    :                119930
  Current MAC count:                     3 (Limit 8192)
Local interface: ae23.0, Index: 1040
  Broadcast packets:                   226
  Broadcast bytes  :                 14464
  Multicast packets:                585668
  Multicast bytes  :             153464476
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                 26552
  Unicast bytes    :               1947627
  Current MAC count:                     7 (Limit 8192)
Local interface: ae20.0, Index: 1039
  Broadcast packets:                   115
  Broadcast bytes  :                  6900
  Multicast packets:                395113
  Multicast bytes  :              61622869
  Flooded packets  :                     0
  Flooded bytes    :                     0
  Unicast packets  :                  1419
  Unicast bytes    :                117924
  Current MAC count:                     4 (Limit 8192)
```

show ethernet-switching statistics aging

Syntax show ethernet-switching statistics aging

Release Information Command introduced in Junos OS Release 9.4 for EX Series switches.

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 1239
- Configuring MAC Table Aging on Switches on page 81

List of Sample Output show ethernet-switching statistics aging on page 1238

Output Fields Table 140 on page 1237 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 140: 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.

NOTE: For the QFX Series, this command is not supported in Enhanced Layer 2 Software (ELS).

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 table on page 1243
- show ethernet-switching interfaces on page 1215
- show ethernet-switching mac-learning-log on page 1229
- show ethernet-switching table on page 1243
- show ethernet-switching interfaces on page 1215
- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104

List of Sample Output

show ethernet-switching statistics mac-learning on page 1240
show ethernet-switching statistics mac-learning detail on page 1241
show ethernet-switching statistics mac-learning interface ge-0/0/28 detail on page 1241
show ethernet-switching statistics mac-learning interface on page 1241
show ethernet-switching statistics mac-learning detail (QFX Series) on page 1241

Output Fields

Table 141 on page 1240 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 141: 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

List of Syntax

Syntax (QFX Series, QFabric, NFX Series and EX4600)

show ethernet-switching table
<brief | detail | extensive | summary>
<interface *interface-name*>
<management-vlan>
<sort-by (*name* | *tag*)>
<vlan *vlan-name*>

Syntax (EX Series)

show ethernet-switching table
<brief | detail | extensive | summary>
<interface *interface-name*>
<management-vlan>
<persistent-mac <interface *interface-name*>>
<sort-by (*name* | *tag*)>
<vlan *vlan-name*>

Syntax (EX Series, MX Series and QFX Series)

show ethernet-switching table
<brief | count | detail | extensive | summary>
<*address*>
<instance *instance-name*>
<interface *interface-name*>
isid *isid*
<logical-system *logical-system-name*>
<persistent-learning (interface *interface-name* | mac *mac-address*)>
<*address*>
<vlan-id (all-vlan | *vlan-id*)>
<vlan-name (all | *vlan-name*)>

Syntax (SRX Series)

show ethernet-switching table (brief |detail |extensive) interface *interface-name*

Release Information

Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 9.5 for SRX Series.
Options **summary**, **management-vlan**, and **vlan *vlan-name*** introduced in Junos OS Release 9.6 for EX Series switches.
Option **sort-by** and field name **tag** introduced in Junos OS Release 10.1 for EX Series switches.
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.
Option **persistent-mac** introduced in Junos OS Release 11.4 for EX Series switches.
Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Options **logical-system**, **persistent-learning**, and **summary** introduced in Junos OS Release 13.2X50-D10 (ELS).

Description Displays the Ethernet switching table.

(MX Series routers,EX Series switches only) Displays Layer 2 MAC address information.

Options For QFX Series, QFabric, NFX Series and EX4600:

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.

persistent-mac <interface *interface-name*>—(Optional) Display the persistent MAC addresses learned for all interfaces or a specified interface. You can use this command to view entries that you want to clear for an interface that you intentionally disabled.

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.

For EX Series, MX Series and QFX Series:

none—Display all learned Layer 2 MAC address information.

brief | count | detail | extensive | summary—(Optional) Display the specified level of output.

address—(Optional) Display the specified learned Layer 2 MAC address information.

instance *instance-name*—(Optional) Display learned Layer 2 MAC addresses for the specified routing instance.

interface *interface-name*—(Optional) Display learned Layer 2 MAC addresses for the specified interface.

isid *isid*—(Optional) Display learned Layer 2 MAC addresses for the specified ISID.

logical-system *logical-system-name*—(Optional) Display Ethernet-switching statistics information for the specified logical system.

persistent-learning (interface *interface-name* | mac *mac-address*)—(Optional) Display dynamically learned MAC addresses that are retained despite device restarts and interface failures for a specified interface, or information about a specified MAC address.

vlan-id (all-vlan | *vlan-id*)—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.

vlan-name (all | *vlan-name*)—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.

For SRX Series:

- **none**—(Optional) Display brief information about the Ethernet switching table.
- **brief | detail | extensive**—(Optional) Display the specified level of output.
- **interface-name**—(Optional) Display the Ethernet switching table for a specific interface.

Additional Information

When Layer 2 protocol tunneling is enabled, the tunneling MAC address 01:00:0c:cd:cd:d0 is installed in the MAC table. When the Cisco Discovery Protocol (CDP), Spanning Tree Protocol (STP), or VLAN Trunk Protocol (VTP) is configured for Layer 2 protocol tunneling on an interface, the corresponding protocol MAC address is installed in the MAC table.

Required Privilege Level

view

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Example: Setting Up Bridging with Multiple VLANs on page 141
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- clear ethernet-switching table on page 1173
- show ethernet-switching mac-learning-log on page 1229

List of Sample Output

show ethernet-switching table (Enhanced Layer 2 Software on QFX Series, QFabric, NFX Series and EX460) on page 1249
show ethernet-switching table (QFX Series, QFabric, NFX Series and EX460) on page 1250
show ethernet-switching table (Private VLANs on QFX Series, QFabric, NFX Series and EX460) on page 1251
show ethernet-switching table (Junos Fusion Data Center with EVPN on QFX Series switches) on page 1251
show ethernet-switching table brief (QFX Series, QFabric, NFX Series and EX460) on page 1252
show ethernet-switching table detail (QFX Series, QFabric, NFX Series and EX460) on page 1253
show ethernet-switching table extensive (QFX Series, QFabric, NFX Series and EX460) on page 1254
show ethernet-switching table interface (QFX Series, QFabric, NFX Series and EX460) on page 1256
show ethernet-switching table (EX Series switches) on page 1256
show ethernet-switching table brief (EX Series switches) on page 1256
show ethernet-switching table detail (EX Series switches) on page 1257

show ethernet-switching table extensive (EX Series switches) on page 1258
show ethernet-switching table persistent-mac (EX Series switches) on page 1258
show ethernet-switching table persistent-mac interface ge-0/0/16.0 (EX Series switches) on page 1258
show ethernet-switching table (EX Series, MX Series and QFX Series) on page 1258
show ethernet-switching table brief on page 1260
show ethernet-switching table count on page 1261
show ethernet-switching table extensive on page 1262
show ethernet-switching table detail (SRX Series) on page 1263
show ethernet-switching table extensive (SRX Series) on page 1264
show ethernet-switching table interface ge-0/0/1 (SRX Series) on page 1265

Output Fields For QFX Series, QFabric, NFX Series and EX4600:

The following table lists the output fields for the **show ethernet-switching table** command on QFX Series, QFabric, NFX Series and EX4600. Output fields are listed in the approximate order in which they appear.

Table 142: show ethernet-switching table Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of a VLAN.	All levels
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**

For EX Series switches:

The following table lists the output fields for the **show ethernet-switching table** command on EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 143: show ethernet-switching table Output Fields

Field Name	Field Description	Level of Output
VLAN	The name of a VLAN.	All levels
Tag	The VLAN ID tag name or number.	**extensive**
MAC or **MAC address**	The MAC address associated with the VLAN.	All levels
Type	The type of MAC address. Values are: • **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. • **persistent**—The learned MAC addresses that will persist across restarts of the switch or interface-down events.	All levels except **persistent-mac**
Type	The type of MAC address. Values are: • **installed**—addresses that are in the Ethernet switching table. • **uninstalled**—addresses that could not be installed in the table or were uninstalled in an interface-down event and will be reinstalled in the table when the interface comes back up.	**persistent-mac**
Age	The 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 **All-members** (flood entry).	All levels
Learned	For learned entries, the time which the entry was added to the Ethernet switching table.	**detail**, **extensive**
Nexthop index	The next-hop index number.	**detail**, **extensive**
persistent-mac	**installed** indicates MAC addresses that are in the Ethernet switching table and **uninstalled** indicates MAC addresses that could not be installed in the table or were uninstalled in an interface-down event (and will be reinstalled in the table when the interface comes back up).	

For EX Series, MX Series and QFX Series:

The table describes the output fields for the **show ethernet-switching table** command on EX Series, MX Series and QFX Series. Output fields are listed in the approximate order in which they appear.

Table 144: show ethernet-switching table Output fields

Field Name	Field Description
Routing instance	Name of the routing instance.
VLAN name	Name of the VLAN.

Table 144: show ethernet-switching table Output fields (continued)

Field Name	Field Description
MAC address	MAC address or addresses learned on a logical interface.
MAC flags	Status of MAC address learning properties for each interface: • **S**—Static MAC address is configured. • **D**—Dynamic MAC address is configured. • **L**—Locally learned MAC address is configured. • **SE**—MAC accounting is enabled. • **NM**—Non-configured MAC. • **R**—Locally learned MAC address is configured.
Age	This field is not supported.
Logical interface	Name of the logical interface.
Active source	IP address of remote entity on which MAC address is learned.
MAC count	Number of MAC addresses learned on the specific routing instance or interface.
Learning interface	Name of the logical interface on which the MAC address was learned.
Learning VLAN	VLAN ID of the routing instance or VLAN in which the MAC address was learned.
Layer 2 flags	Debugging flags signifying that the MAC address is present in various lists.
Epoch	Spanning-tree-protocolepoch number identifying when the MAC address was learned. Used for debugging.
Sequence number	Sequence number assigned to this MAC address. Used for debugging.
Learning mask	Mask of the Packet Forwarding Engines where this MAC address was learned. Used for debugging.
IPC generation	Creation time of the logical interface when this MAC address was learned. Used for debugging.

For SRX Series:

Table 145 on page 1248 lists the output fields for the **show ethernet-switching table** command. Output fields are listed in the approximate order in which they appear.

Table 145: show ethernet-switching table Output Fields

Field Name	Field Description
VLAN	The name of a VLAN.

Table 145: show ethernet-switching table Output Fields (continued)

Field Name	Field Description
MAC address	The MAC address associated with the VLAN.
Type	The type of MAC address. Values are: • 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.
Age	The time remaining before the entry ages out and is removed from the Ethernet switching table.
Interfaces	Interface associated with learned MAC addresses or All-members (flood entry).
Learned	For learned entries, the time which the entry was added to the Ethernet switching table.

Sample Output

show ethernet-switching table (Enhanced Layer 2 Software on QFX Series, QFabric, NFX Series and EX460)

```
user@switch> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
O - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan1               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan1               b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
O - ovsdb MAC)

Ethernet switching table : 2 entries, 2 learned

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan10              b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan10              b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
```

```
O - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan2               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan2               b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
O - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan3               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan3               b0:c6:9a:ca:3c:03   D             -    ae1.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
O - ovsdb MAC)


Ethernet switching table : 2 entries, 2 learned
Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    vlan4               b0:c6:9a:ca:3c:01   D             -    ae1.0

    vlan4               b0:c6:9a:ca:3c:03   D             -    ae1.0
```

show ethernet-switching table (QFX Series, QFabric, NFX Series and EX460)

```
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 on QFX Series, QFabric, NFX Series and EX460)

```
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 (Junos Fusion Data Center with EVPN on QFX Series switches)

```
user@switch> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent
static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC,
O - ovsdb MAC)

Ethernet switching table : 30 entries, 30 learned
Routing instance : default-switch
   Vlan                MAC                 MAC      Logical                 Active

   name                address             flags    interface               source

   v100                00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v100                7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v100                7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v101                00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v101                7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v101                7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v102                00:31:46:e8:f9:d6   D        vtep.32768
```

```
192.168.2.22
   v102                7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v102                7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v103                00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v103                7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v103                7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v3001               00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v3001               28:c0:da:6a:9f:c2   DL       ae11.0
   v3001               7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v3001               7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v3002               00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v3002               7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v3002               7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v3003               00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v3003               28:c0:da:6a:9f:c2   DL       ae11.0
   v3003               7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v3003               7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v3004               00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v3004               7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v3004               7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
   v3005               00:31:46:e8:f9:d6   D        vtep.32768
192.168.2.22
   v3005               28:c0:da:6a:9f:c2   DL       ae11.0
   v3005               7c:e2:ca:e2:75:7c   D        vtep.32771
192.168.4.44
   v3005               7c:e2:ca:e4:05:9a   D        vtep.32770
192.168.3.33
```

show ethernet-switching table brief (QFX Series, QFabric, NFX Series and EX460)

```
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 (QFX Series, QFabric, NFX Series and EX460)

```
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 (QFX Series, QFabric, NFX Series and EX460)

```
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 (QFX Series, QFabric, NFX Series and EX460)

```
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 ethernet-switching table (EX Series switches)

```
user@switch> show ethernet-switching table
Ethernet-switching table: 57 entries, 15 learned, 2 persistent
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 ge-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 ge-0/0/47.0
  T1                *                 Flood          - All-members
  T1                00:00:05:00:00:01 Persistent     0 ge-0/0/46.0
  T1                00:00:5e:00:01:00 Static         - Router
  T1                00:19:e2:50:63:e0 Persistent     0 ge-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 ge-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 ge-0/0/15.0
  T111              00:19:e2:50:7d:e0 Static         - Router
  T111              00:19:e2:50:ac:00 Learn          0 ge-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 ge-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 ge-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 ge-0/0/46.0
[output truncated]
```

show ethernet-switching table brief (EX Series switches)

```
user@switch> show ethernet-switching table brief
Ethernet-switching table: 57 entries, 15 learned, 2 persistent entries
  VLAN              MAC address       Type         Age Interfaces
  F2                *                 Flood          - All-members
  F2                00:00:05:00:00:03 Learn          0 ge-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 ge-0/0/47.0
  T1                  *                 Flood         - All-members
  T1                  00:00:05:00:00:01 Persistent    0 ge-0/0/46.0
  T1                  00:00:5e:00:01:00 Static        - Router
  T1                  00:19:e2:50:63:e0 Persistent    0 ge-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 ge-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 ge-0/0/15.0
  T111                00:19:e2:50:7d:e0 Static        - Router
  T111                00:19:e2:50:ac:00 Learn         0 ge-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 ge-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 ge-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 ge-0/0/46.0
[output truncated]
```

show ethernet-switching table detail (EX Series switches)

```
user@switch> show ethernet-switching table detail
Ethernet-switching table: 5 entries, 2 learned entries
 VLAN: default, Tag: 0, MAC: *, Interface: All-members
  Interfaces:
   ge-0/0/11.0, ge-0/0/20.0, ge-0/0/30.0, ge-0/0/36.0, ge-0/0/3.0
  Type: Flood
  Nexthop index: 1307

 VLAN: default, Tag: 0, MAC: 00:1f:12:30:b8:83, Interface: ge-0/0/3.0
   Type: Learn, Age: 0, Learned: 20:09:26
   Nexthop index: 1315

 VLAN: v1, Tag: 101, MAC: *, Interface: All-members
  Interfaces:
   ge-0/0/31.0
  Type: Flood
  Nexthop index: 1313

 VLAN: v1, Tag: 101, MAC: 00:1f:12:30:b8:89, Interface: ge-0/0/31.0
  Type: Learn, Age: 0, Learned: 20:09:25
  Nexthop index: 1312

 VLAN: v2, Tag: 102, MAC: *, Interface: All-members
  Interfaces:
   ae0.0
  Type: Flood
  Nexthop index: 1317
```

show ethernet-switching table extensive (EX Series switches)

```
user@switch> show ethernet-switching table extensive
Ethernet-switching table: 3 entries, 1 learned, 5 persistent entries

    VLAN: v1, Tag: 10, MAC: *, Interface: All-members
    Interfaces:
               ge-0/0/14.0, 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-0/0/10.0,
               ge-0/0/0.0
    Type: Flood
    Nexthop index: 567

    VLAN: v1, Tag: 10, MAC: 00:21:59:c6:93:22, Interface: Router
    Type: Static
    Nexthop index: 0

    VLAN: v1, Tag: 10, MAC: 00:21:59:c9:9a:4e, Interface: ge-0/0/14.0
    Type: Learn, Age: 0, Learned: 18:40:50
    Nexthop index: 564
```

show ethernet-switching table persistent-mac (EX Series switches)

```
user@switch> show ethernet-switching table persistent-mac
VLAN              MAC address       Type       Interface
default           00:10:94:00:00:02 installed    ge-0/0/42.0
default           00:10:94:00:00:03 installed    ge-0/0/42.0
default           00:10:94:00:00:04 installed    ge-0/0/42.0
default           00:10:94:00:00:05 installed    ge-0/0/42.0
default           00:10:94:00:00:06 installed    ge-0/0/42.0
default           00:10:94:00:05:02 uninstalled ge-0/0/16.0
default           00:10:94:00:06:03 uninstalled ge-0/0/16.0
default           00:10:94:00:07:04 uninstalled ge-0/0/16.0
```

show ethernet-switching table persistent-mac interface ge-0/0/16.0 (EX Series switches)

```
VLAN              MAC address       Type       Interface
default           00:10:94:00:05:02 uninstalled ge-0/0/16.0
default           00:10:94:00:06:03 uninstalled ge-0/0/16.0
default           00:10:94:00:07:04 uninstalled ge-0/0/16.0
```

show ethernet-switching table (EX Series, MX Series and QFX Series)

```
user@host> show ethernet-switching table
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN101             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN102             88:e0:f3:bb:07:f0   D             -    ae20.0
```

```
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN103             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN104             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN1101            00:1f:12:32:f5:c1   D             -    ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN1102            00:1f:12:32:f5:c1   D             -    ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN1103            00:1f:12:32:f5:c1   D             -    ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN1104            00:1f:12:32:f5:c1   D             -    ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
```

```
    name                address             flags              interface
    VLAN1105            00:1f:12:32:f5:c1   D             -    ae0.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN1106            00:1f:12:32:f5:c1   D             -    ae0.0
[...output truncated...]
```

show ethernet-switching table brief

```
user@host> show ethernet-switching table brief
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN101             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN102             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN103             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN104             88:e0:f3:bb:07:f0   D             -    ae20.0


MAC flags (S - static MAC, D - dynamic MAC, L - locally learned
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC)

Routing instance : default-switch
    Vlan                MAC                 MAC         Age    Logical
    name                address             flags              interface
    VLAN1101            00:1f:12:32:f5:c1   D             -    ae0.0
[...output truncated...]
```

show ethernet-switching table count

```
user@host> show ethernet-switching table count
0 MAC address learned in routing instance default-switch VLAN VLAN1000
ae26.0:1000

1 MAC address learned in routing instance default-switch VLAN VLAN101
ae20.0:101

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           101              1                0

1 MAC address learned in routing instance default-switch VLAN VLAN102
ae20.0:102

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           102              1                0

1 MAC address learned in routing instance default-switch VLAN VLAN103
ae20.0:103

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           103              1                0

1 MAC address learned in routing instance default-switch VLAN VLAN104
ae20.0:104

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
           104              1                0

0 MAC address learned in routing instance default-switch VLAN VLAN105
ae20.0:105

0 MAC address learned in routing instance default-switch VLAN VLAN106
ae20.0:106

0 MAC address learned in routing instance default-switch VLAN VLAN107
ae20.0:107

0 MAC address learned in routing instance default-switch VLAN VLAN108
ae20.0:108

0 MAC address learned in routing instance default-switch VLAN VLAN109
ae20.0:109

0 MAC address learned in routing instance default-switch VLAN VLAN110
ae20.0:110

1 MAC address learned in routing instance default-switch VLAN VLAN1101
ae0.0:1101

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
          1101              1                0

1 MAC address learned in routing instance default-switch VLAN VLAN1102
```

```
ae0.0:1102

  MAC address count per learn VLAN within routing instance:
    Learn VLAN ID      MAC count      Static MAC count
          1102              1                 0
[...output truncated...]
```

show ethernet-switching table extensive

```
user@host> show ethernet-switching table extensive

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 101
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                            Sequence number: 2
   Learning mask: 0x00000008

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 102
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                            Sequence number: 2
   Learning mask: 0x00000008

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 103
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                            Sequence number: 2
   Learning mask: 0x00000008

MAC address: 88:e0:f3:bb:07:f0
  Routing instance: default-switch
VLAN ID: 104
   Learning interface: ae20.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                            Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1101
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                            Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1102
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                            Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
```



```
VLAN ID: 1103
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008

MAC address: 00:1f:12:32:f5:c1
  Routing instance: default-switch
VLAN ID: 1104
   Learning interface: ae0.0
   Layer 2 flags: in_hash,in_ifd,in_ifl,in_vlan,in_rtt,kernel,in_ifbd
   Epoch: 0                          Sequence number: 2
   Learning mask: 0x00000008
```

Sample Output

show ethernet-switching table detail (SRX Series)

```
user@host> show ethernet-switching table detail
Ethernet-switching table: 57 entries, 17 learned
F2, *
Interface(s): ge-0/0/44.0
Type: Flood
F2, 00:00:5E:00:53:AC
Interface(s): ge-0/0/44.0
Type: Learn, Age: 0, Learned: 2:03:09
F2, 00:00:5E:00:53:AA
Interface(s): Router
Type: Static
Linux, *
Interface(s): ge-0/0/47.0
Type: Flood
Linux, 00:00:5E:00:53:AB
Interface(s): Router
Type: Static
Linux, 00:00:5E:00:53:AC
Interface(s): ge-0/0/47.0
Type: Learn, Age: 0, Learned: 2:03:08
T1, *
Interface(s): ge-0/0/46.0
Type: Flood
T1, 00:00:5E:00:53:AD
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AE
Interface(s): Router
Type: Static
T1, 00:00:5E:00:53:AF
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AG
Interface(s): Router
Type: Static
T10, *
Interface(s): ge-0/0/46.0
Type: Flood
T10, 00:00:5E:00:53:AH
Interface(s): Router
Type: Static
T10, 00:00:5E:00:53:AI
Interface(s): ge-0/0/46.0
```

```
Type: Learn, Age: 0, Learned: 2:03:08
T10, 00:00:5E:00:53:AJ
Interface(s): Router
Type: Static
T111, *
Interface(s): ge-0/0/15.0
Type: Flood
[output truncated]
```

Sample Output

show ethernet-switching table extensive (SRX Series)

```
user@host> show ethernet-switching table extensive
Ethernet-switching table: 57 entries, 17 learned
F2, *
Interface(s): ge-0/0/44.0
Type: Flood
F2, 00:00:5E:00:53:AC
Interface(s): ge-0/0/44.0
Type: Learn, Age: 0, Learned: 2:03:09
F2, 00:00:5E:00:53:AA
Interface(s): Router
Type: Static
Linux, *
Interface(s): ge-0/0/47.0
Type: Flood
Linux, 00:00:5E:00:53:AB
Interface(s): Router
Type: Static
Linux, 00:00:5E:00:53:AC
Interface(s): ge-0/0/47.0
Type: Learn, Age: 0, Learned: 2:03:08
T1, *
Interface(s): ge-0/0/46.0
Type: Flood
T1, 00:00:5E:00:53:AD
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AE
Interface(s): Router
Type: Static
T1, 00:00:5E:00:53:AF
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:07
T1, 00:00:5E:00:53:AG
Interface(s): Router
Type: Static
T10, *
Interface(s): ge-0/0/46.0
Type: Flood
T10, 00:00:5E:00:53:AH
Interface(s): Router
Type: Static
T10, 00:00:5E:00:53:AI
Interface(s): ge-0/0/46.0
Type: Learn, Age: 0, Learned: 2:03:08
T10, 00:00:5E:00:53:AJ
Interface(s): Router
Type: Static
T111, *
```

```
Interface(s): ge-0/0/15.0
Type: Flood
[output truncated]
```

Sample Output

show ethernet-switching table interface ge-0/0/1 (SRX Series)

```
user@host> show ethernet-switching table interface ge-0/0/1
Ethernet-switching table: 1 unicast entries
VLAN     MAC address         Type     Age Interfaces
V1       *                   Flood    - All-members
V1       00:00:5E:00:53:AF   Learn    0 ge-0/0/1.0
```

show interfaces

List of Syntax

Syntax (Gigabit Ethernet) on page 1266
Syntax (10 Gigabit Ethernet) on page 1266
Syntax (SRX Series Devices) on page 1266

Syntax (Gigabit Ethernet)

```
show interfaces ge-fpc/pic/port
<brief | detail | extensive | terse>
<descriptions>
<media>
<snmp-index snmp-index>
<statistics>
```

Syntax (10 Gigabit Ethernet)

```
show interfaces xe-fpc/pic/port
<brief | detail | extensive | terse>
<descriptions>
<media>
<snmp-index snmp-index>
<statistics>
```

Syntax (SRX Series Devices)

```
show interfaces (
  <interface-name>
  <brief | detail | extensive | terse>
  <controller interface-name>|
  <descriptions interface-name>|
  <destination-class (all | destination-class-name logical-interface-name)>|
  <diagnostics optics interface-name>|
  <far-end-interval interface-fpc/pic/port>|
  <filters interface-name>|
  <flow-statistics interface-name>|
  <interval interface-name>|
  <load-balancing (detail | interface-name)>|
  <mac-database mac-address mac-address>|
  <mc-ae id identifier unit number revertive-info>|
  <media interface-name>|
  <policers interface-name>|
  <queue both-ingress-egress egress forwarding-class forwarding-class ingress l2-statistics>|
  <redundancy (detail | interface-name)>|
  <routing brief detail summary interface-name>|
  <routing-instance (all | instance-name)>|
  <snmp-index snmp-index>|
  <source-class (all | destination-class-name logical-interface-name)>|
  <statistics interface-name>|
  <switch-port switch-port number>|
  <transport pm (all | optics | otn) (all | current | currentday | interval | previousday) (all |
    interface-name)>|
  <zone interface-name>
)
```

Release Information

Command introduced before Junos OS Release 7.4 for Gigabit interfaces.
Command introduced in Junos OS Release 8.0 for 10 Gigabit interfaces.
Command modified in Junos OS Release 9.5 for SRX Series devices.

Command introduced in Junos OS Release 18.1 for Gigabit interfaces.

Description Display status information about the specified Gigabit Ethernet interface.

(M320, M120, MX Series, and T Series routers only) Display status information about the specified 10-Gigabit Ethernet interface.

Display the IPv6 interface traffic statistics about the specified Gigabit Ethernet interface for MX series routers. The input and output bytes (bps) and packets (pps) rates are not displayed for IFD and local traffic.

Display status information and statistics about interfaces on SRX Series appliance running Junos OS.

NOTE: On SRX Series appliances, on configuring identical IPs on a single interface, you will not see a warning message; instead, you will see a syslog message.

Options For Gigabit interfaces:

ge-*fpc/pic/port*—Display standard information about the specified Gigabit Ethernet interface.

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.

snmp-index *snmp-index*—(Optional) Display information for the specified SNMP index of the interface.

statistics—(Optional) Display static interface statistics.

For 10 Gigabit interfaces:

xe-*fpc/pic/port*—Display standard information about the specified 10-Gigabit Ethernet interface.

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.

snmp-index *snmp-index*—(Optional) Display information for the specified SNMP index of the interface.

statistics—(Optional) Display static interface statistics.

For SRX interfaces:

- **interface-name**—(Optional) Display standard information about the specified interface. Following is a list of typical interface names. Replace pim with the PIM slot and port with the port number.
  - **at-** ***pim*/0/*port***—ATM-over-ADSL or ATM-over-SHDSL interface.
  - **ce1-*pim*/0/ *port***—Channelized E1 interface.
  - **cl-0/0/8**—3G wireless modem interface for SRX320 devices.
  - **ct1-*pim*/0/*port***—Channelized T1 interface.
  - **dl0**—Dialer Interface for initiating ISDN and USB modem connections.
  - **e1-*pim*/0/*port***—E1 interface.
  - **e3-*pim*/0/*port***—E3 interface.
  - **fe-*pim*/0/*port***—Fast Ethernet interface.
  - **ge-*pim*/0/*port***—Gigabit Ethernet interface.
  - **se-*pim*/0/*port***—Serial interface.
  - **t1-*pim*/0/*port***—T1 (also called DS1) interface.
  - **t3-*pim*/0/*port***—T3 (also called DS3) interface.
  - **wx-*slot*/0/0**—WAN acceleration interface, for the WXC Integrated Services Module (ISM 200).
  - **interface-name**—(Optional) Display standard information about the specified interface. Following is a list of typical interface names. Replace pim with the PIM slot and port with the port number.
    - **at-** ***pim*/0/*port***—ATM-over-ADSL or ATM-over-SHDSL interface.
    - **ce1-*pim*/0/ *port***—Channelized E1 interface.
    - **cl-0/0/8**—3G wireless modem interface for SRX320 devices.
    - **ct1-*pim*/0/*port***—Channelized T1 interface.
    - **dl0**—Dialer Interface for initiating ISDN and USB modem connections.
    - **e1-*pim*/0/*port***—E1 interface.
    - **e3-*pim*/0/*port***—E3 interface.
    - **fe-*pim*/0/*port***—Fast Ethernet interface.
    - **ge-*pim*/0/*port***—Gigabit Ethernet interface.
    - **se-*pim*/0/*port***—Serial interface.
    - **t1-*pim*/0/*port***—T1 (also called DS1) interface.

- **t3-*pim*/0/*port***—T3 (also called DS3) interface.
- **wx-*slot*/0/0**—WAN acceleration interface, for the WXC Integrated Services Module (ISM 200).

Additional Information

In a logical system, this command displays information only about the logical interfaces and not about the physical interfaces.

Required Privilege Level

view

Related Documentation

- Understanding Layer 2 Interfaces on Security Devices on page 637
- *Verifying and Managing Agent Circuit Identifier-Based Dynamic VLAN Configuration*
- *Verifying and Managing Configurations for Dynamic VLANs Based on Access-Line Identifiers*

List of Sample Output

Output Fields Table 146 on page 1270 describes the output fields for the **show interfaces** (Gigabit Ethernet) command. Output fields are listed in the approximate order in which they appear. For Gigabit Ethernet IQ and IQE PICs, the traffic and MAC statistics vary by interface type. For more information, see Table 147 on page 1298.

Table 146: show interfaces (Gigabit Ethernet) 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. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.	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

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
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
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
WAN-PHY mode	10-Gigabit Ethernet interface operating in Wide Area Network Physical Layer Device (WAN PHY) mode. WAN PHY allows 10-Gigabit Ethernet wide area links to use fiber-optic cables and other devices intended for SONET/SDH.	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**.	All levels
Auto-negotiation	(Gigabit Ethernet interfaces) Autonegotiation status: **Enabled** or **Disabled**.	All levels
Remote-fault	(Gigabit Ethernet interfaces) 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. Possible values are described in the "Device Flags" section under *Common Output Fields Description*.	All levels
Interface flags	Information about the interface. Possible values are described in the "Interface Flags" section under *Common Output Fields Description*.	All levels
Link flags	Information about the link. Possible values are described in the "Links Flags" section under *Common Output Fields Description*.	All levels
Wavelength	(10-Gigabit Ethernet dense wavelength-division multiplexing [DWDM] interfaces) Displays the configured wavelength, in nanometers (nm).	All levels
Frequency	(10-Gigabit Ethernet DWDM interfaces only) Displays the frequency associated with the configured wavelength, in terahertz (THz).	All levels
CoS queues	Number of CoS queues configured.	**detail extensive** none

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
Schedulers	(Gigabit Ethernet intelligent queuing 2 [IQ2] interfaces only) Number of CoS schedulers configured.	**extensive**
Hold-times	Current interface hold-time up and hold-time down, in milliseconds (ms).	**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: 2002-04-26 10:52:40 PDT (04:33:20 ago).**	**detail extensive** none
Input Rate	Input rate in bits per second (bps) and packets per second (pps). The value in this field also includes the Layer 2 overhead bytes for ingress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.	None
Output Rate	Output rate in bps and pps. The value in this field also includes the Layer 2 overhead bytes for egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.	None
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**
Egress account overhead	Layer 2 overhead in bytes that is accounted in the interface statistics for egress traffic.	**detail extensive**
Ingress account overhead	Layer 2 overhead in bytes that is accounted in the interface statistics for ingress traffic.	**detail extensive**
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. The value in this field also includes the Layer 2 overhead bytes for ingress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Output bytes**—Number of bytes transmitted on the interface. The value in this field also includes the Layer 2 overhead bytes for egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface. Gigabit Ethernet and 10-Gigabit Ethernet IQ PICs count the overhead and CRC bytes. For Gigabit Ethernet IQ PICs, the input byte counts vary by interface type. For more information, see Table 31 under the show interfaces command.	**detail extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
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 (usually IPv4) 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 by configuring 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 146: show interfaces (Gigabit Ethernet) 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. NOTE: Due to accounting space limitations on certain Type 3 FPCs (which are supported in M320 and T640 routers), the **Drops** field does not always use the correct value for queue 6 or queue 7 for interfaces on 10-port 1-Gigabit Ethernet PICs. • **Collisions**—Number of Ethernet collisions. The Gigabit Ethernet PIC supports only full-duplex operation, so for Gigabit Ethernet PICs, this number must always be 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 must 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. NOTE: In DPCs that are not of the enhanced type, such as DPC 40x 1GE R, DPCE 20x 1GE + 2x 10GE R, or DPCE 40x 1GE R, you might notice a discrepancy in the output of the **show interfaces** command because incoming packets might be counted in the Egress queues section of the output. This problem occurs on non-enhanced DPCs because the egress queue statistics are polled from IMQ (Inbound Message Queuing) block of the I-chip. The IMQ block does not differentiate between ingress and egress WAN traffic; as a result, the combined statistics are displayed in the egress queue counters on the Routing Engine. In a simple VPLS scenario, if there is no MAC entry in DMAC table (by sending unidirectional traffic), traffic is flooded and the input traffic is accounted in IMQ. For bidirectional traffic (MAC entry in DMAC table), if the outgoing interface is on the same I-chip then both ingress and egress statistics are counted in a combined way. If the outgoing interface is on a different I-chip or FPC, then only egress statistics are accounted in IMQ. This behavior is expected with non-enhanced DPCs	**detail extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
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. NOTE: Due to accounting space limitations on certain Type 3 FPCs (which are supported in M320 and T640 routers), the **Dropped packets** field does not always display the correct value for queue 6 or queue 7 for interfaces on 10-port 1-Gigabit Ethernet PICs.	**detail extensive**
Ingress queues	Total number of ingress queues supported on the specified interface. Displayed on IQ2 interfaces.	**extensive**
Queue counters (Ingress)	CoS queue number and its associated user-configured forwarding class name. Displayed on IQ2 interfaces. • **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**
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 router configuration, an alarm can ring the red or yellow alarm bell on the router, 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
Interface transmit statistics	(On MX Series devices) Status of the **interface-transmit-statistics** configuration: Enabled or Disabled. • **Enabled**—When the **interface-transmit-statistics** statement is included in the configuration. If this is configured, the interface statistics show the actual transmitted load on the interface. • **Disabled**—When the **interface-transmit-statistics** statement is not included in the configuration. If this is not configured, the interface statistics show the offered load on the interface.	**detail extensive**
OTN FEC statistics	The forward error correction (FEC) counters provide the following statistics: • **Corrected Errors**—Count of corrected errors in the last second. • **Corrected Error Ratio**—Corrected error ratio in the last 25 seconds. For example, 1e-7 is 1 error per 10 million bits.	**detail extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
PCS statistics	(10-Gigabit Ethernet interfaces) Displays Physical Coding Sublayer (PCS) fault conditions from the WAN PHY or the LAN PHY device. • **Bit errors**—Number of seconds during which at least one bit error rate (BER) occurred while the PCS receiver is operating in normal mode. • **Errored blocks**—Number of seconds when at least one errored block occurred while the PCS receiver is operating in normal mode.	**detail extensive**

Link Degrade	Shows the link degrade status of the physical link and the estimated bit error rates (BERs). This field is available only for the PICs supporting the physical link monitoring feature.
	• **Link Monitoring**—Indicates if physical link degrade monitoring is enabled on the interface.
	  • **Enable**—Indicates that link degrade monitoring has been enabled (using the **link-degrade-monitor** statement) on the interface.
	  • **Disable**—Indicates that link degrade monitoring has not been enabled on the interface. If link degrade monitoring has not been enabled, the output does not show any related information, such as BER values and thresholds.
	• **Link Degrade Set Threshold**—The BER threshold value at which the link is considered degraded and a corrective action is triggered.
	• **Link Degrade Clear Threshold**—The BER threshold value at which the degraded link is considered recovered and the corrective action applied to the interface is reverted.
	• **Estimated BER**—The estimated bit error rate.
	• **Link-degrade event**—Shows link degrade event information.
	  • **Seconds**—Time (in seconds) elapsed after a link degrade event occurred.
	  • **Count**—The number of link degrade events recorded.
	  • **State**—Shows the link degrade status (example: **Defect Active**).
		detail extensive

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
MAC statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC subsystem, including the following: • **Total octets** and **total packets**—Total number of octets and packets. For Gigabit Ethernet IQ PICs, the received octets count varies by interface type. For more information, see Table 31 under the show interfaces command. • **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 or a cable 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**—There are two possible conditions regarding the number of oversized frames: ◦ Packet length exceeds 1518 octets, or ◦ Packet length exceeds MRU • **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. NOTE: The 20-port Gigabit Ethernet MIC (MIC-3D-20GE-SFP) does not have hardware counters for VLAN frames. Therefore, the **VLAN tagged frames** field displays 0 when the **show interfaces** command is executed on a 20-port Gigabit Ethernet MIC. In other words, the number of VLAN tagged frames cannot be determined for the 20-port Gigabit Ethernet MIC. • **Code violations**—Number of times an event caused the PHY to indicate “Data reception error” or “invalid data symbol error.”	**extensive**
OTN Received Overhead Bytes	APS/PCC0: 0x02, APS/PCC1: 0x11, APS/PCC2: 0x47, APS/PCC3: 0x58 Payload Type: 0x08	**extensive**
OTN Transmitted Overhead Bytes	APS/PCC0: 0x00, APS/PCC1: 0x00, APS/PCC2: 0x00, APS/PCC3: 0x00 Payload Type: 0x08	**extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
Filter statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC address filter subsystem. The filtering is done by the content-addressable memory (CAM) on the PIC. The filter examines a packet's source and destination MAC addresses to determine whether the packet may enter the system or be rejected. • **Input packet count**—Number of packets received from the MAC hardware that the filter processed. • **Input packet rejects**—Number of packets that the filter rejected because of either the source MAC address or the destination MAC address. • **Input DA rejects**—Number of packets that the filter rejected because the destination MAC address of the packet is not on the accept list. It is normal for this value to increment. When it increments very quickly and no traffic is entering the router from the far-end system, either there is a bad ARP entry on the far-end system, or multicast routing is not on and the far-end system is sending many multicast packets to the local router (which the router is rejecting). • **Input SA rejects**—Number of packets that the filter rejected because the source MAC address of the packet is not on the accept list. The value in this field must increment only if source MAC address filtering has been enabled. If filtering is enabled, if the value increments quickly, and if the system is not receiving traffic that it should from the far-end system, it means that the user-configured source MAC addresses for this interface are incorrect. • **Output packet count**—Number of packets that the filter has given to the MAC hardware. • **Output packet pad count**—Number of packets the filter padded to the minimum Ethernet size (60 bytes) before giving the packet to the MAC hardware. Usually, padding is done only on small ARP packets, but some very small IP packets can also require padding. If this value increments rapidly, either the system is trying to find an ARP entry for a far-end system that does not exist or it is misconfigured. • **Output packet error count**—Number of packets with an indicated error that the filter was given to transmit. These packets are usually aged packets or are the result of a bandwidth problem on the FPC hardware. On a normal system, the value of this field must not increment. • **CAM destination filters, CAM source filters**—Number of entries in the CAM dedicated to destination and source MAC address filters. There can only be up to 64 source entries. If source filtering is disabled, which is the default, the values for these fields must be 0.	**extensive**
PMA PHY	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET error information: • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem. Subfields are: • **PHY Lock**—Phase-locked loop • **PHY Light**—Loss of optical signal	**extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
WIS section	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET error information: • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem. Subfields are: • **BIP-B1**—Bit interleaved parity for SONET section overhead • **SEF**—Severely errored framing • **LOL**—Loss of light • **LOF**—Loss of frame • **ES-S**—Errored seconds (section) • **SES-S**—Severely errored seconds (section) • **SEFS-S**—Severely errored framing seconds (section)	**extensive**
WIS line	(10-Gigabit Ethernet interfaces, WAN PHY mode) Active alarms and defects, plus counts of specific SONET errors with detailed information: • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem. Subfields are: • **BIP-B2**—Bit interleaved parity for SONET line overhead • **REI-L**—Remote error indication (near-end line) • **RDI-L**—Remote defect indication (near-end line) • **AIS-L**—Alarm indication signal (near-end line) • **BERR-SF**—Bit error rate fault (signal failure) • **BERR-SD**—Bit error rate defect (signal degradation) • **ES-L**—Errored seconds (near-end line) • **SES-L**—Severely errored seconds (near-end line) • **UAS-L**—Unavailable seconds (near-end line) • **ES-LFE**—Errored seconds (far-end line) • **SES-LFE**—Severely errored seconds (far-end line) • **UAS-LFE**—Unavailable seconds (far-end line)	**extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
WIS path	(10-Gigabit Ethernet interfaces, WAN PHY mode) Active alarms and defects, plus counts of specific SONET errors with detailed information: • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem. Subfields are: • **BIP-B3**—Bit interleaved parity for SONET section overhead • **REI-P**—Remote error indication • **LOP-P**—Loss of pointer (path) • **AIS-P**—Path alarm indication signal • **RDI-P**—Path remote defect indication • **UNEQ-P**—Path unequipped • **PLM-P**—Path payload (signal) label mismatch • **ES-P**—Errored seconds (near-end STS path) • **SES-P**—Severely errored seconds (near-end STS path) • **UAS-P**—Unavailable seconds (near-end STS path) • **SES-PFE**—Severely errored seconds (far-end STS path) • **UAS-PFE**—Unavailable seconds (far-end STS path)	**extensive**

Table 146: show interfaces (Gigabit Ethernet) 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 Ethernet interface is connected to a device that performs autonegotiation and the autonegotiation process is successful.
	• **Link partner**—Information from the remote Ethernet device:
	  • **Link mode**—Depending on the capability of the link partner, either **Full-duplex** or **Half-duplex**.
	  • **Flow control**—Types of flow control supported by the link partner. For Gigabit Ethernet interfaces, types are **Symmetric** (link partner supports **PAUSE** on receive and transmit), **Asymmetric** (link partner supports **PAUSE** on transmit), **Symmetric/Asymmetric** (link partner supports **PAUSE** on receive and transmit or only **PAUSE** on transmit), and **None** (link partner does not support flow control).
	  • **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**—Information from the local Ethernet device:
	  • **Flow control**—Types of flow control supported by the local device. For Gigabit Ethernet interfaces, advertised capabilities are **Symmetric/Asymmetric** (local device supports **PAUSE** on receive and transmit or only **PAUSE** on receive) and **None** (local device does not support flow control). Depending on the result of the negotiation with the link partner, local resolution flow control type will display **Symmetric** (local device supports **PAUSE** on receive and transmit), **Asymmetric** (local device supports **PAUSE** on receive), and **None** (local device does not support flow control).
	  • **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

Received path trace, Transmitted path trace	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET/SDH interfaces allow path trace bytes to be sent inband across the SONET/SDH link. Juniper Networks and other router manufacturers use these bytes to help diagnose misconfigurations and network errors by setting the transmitted path trace message so that it contains the system hostname and name of the physical interface. The received path trace value is the message received from the router at the other end of the fiber. The transmitted path trace value is the message that this router transmits.	**extensive**
Packet Forwarding Engine configuration	Information about the configuration of the Packet Forwarding Engine: • **Destination slot**—FPC slot number.	**extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
CoS information	Information about the CoS queue for the physical interface. • **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. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	All levels
VLAN-Tag	Rewrite profile applied to incoming or outgoing frames on the outer (**Out**) VLAN tag or for both the outer and inner (**In**) VLAN tags. • **push**—An outer VLAN tag is pushed in front of the existing VLAN tag. • **pop**—The outer VLAN tag of the incoming frame is removed. • **swap**—The outer VLAN tag of the incoming frame is overwritten with the user-specified VLAN tag information. • **push**—An outer VLAN tag is pushed in front of the existing VLAN tag. • **push-push**—Two VLAN tags are pushed in from the incoming frame. • **swap-push**—The outer VLAN tag of the incoming frame is replaced by a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame. • **swap-swap**—Both the inner and the outer VLAN tags of the incoming frame are replaced by the user-specified VLAN tag value. • **pop-swap**—The outer VLAN tag of the incoming frame is removed, and the inner VLAN tag of the incoming frame is replaced by the user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame. • **pop-pop**—Both the outer and inner VLAN tags of the incoming frame are removed.	**brief detail extensive** none

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
Demux	IP demultiplexing (demux) value that appears if this interface is used as the demux underlying interface. The output is one of the following: • Source Family Inet • Destination Family Inet	**detail extensive** none
Encapsulation	Encapsulation on the logical interface.	All levels
ACI VLAN	Information displayed for agent circuit identifier (ACI) interface set configured with the **agent-circuit-id** autoconfiguration stanza. **Dynamic Profile**—Name of the dynamic profile that defines the ACI interface set. If configured, the ACI interface set enables the underlying Ethernet interface to create dynamic VLAN subscriber interfaces based on ACI information. NOTE: The ACI VLAN field is replaced with the Line Identity field when an ALI interface set is configured with the **line-identity** autoconfiguration stanza.	**brief detail extensive** none
Line Identity	Information displayed for access-line-identifier (ALI) interface sets configured with the **line-identity** autoconfiguration stanza. • **Dynamic Profile**—Name of the dynamic profile that defines the ALI interface set. • Trusted option used to create the ALI interface set: **Circuit-id**, **Remote-id**, or **Accept-no-ids**. More than one option can be configured. If configured, the ALI interface set enables the underlying Ethernet interface to create dynamic VLAN subscriber interfaces based on ALI information. NOTE: The Line Identity field is replaced with the ACI VLAN field when an ACI interface set is configured with the **agent-circuit-id** autoconfiguration stanza.	**detail**
Protocol	Protocol family. Possible values are described in the "Protocol Field" section under *Common Output Fields Description*.	**detail extensive** none
MTU	Maximum transmission unit size on the logical interface.	**detail extensive** none
Neighbor Discovery Protocol (NDP)Queue Statistics	NDP statistics for protocol **inet6** under logical interface statistics. • **Max nh cache**—Maximum interface neighbor discovery nexthop cache size. • **New hold nh limit**—Maximum number of new unresolved nexthops. • **Curr nh cnt**—Current number of resolved nexthops in the NDP queue. • **Curr new hold cnt**—Current number of unresolved nexthops in the NDP queue. • **NH drop cnt**—Number of NDP requests not serviced.	All levels
Dynamic Profile	Name of the dynamic profile that was used to create this interface configured with a Point-to-Point Protocol over Ethernet (PPPoE) family.	**detail extensive** none
Service Name Table	Name of the service name table for the interface configured with a PPPoE family.	**detail extensive** none

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
Max Sessions	Maximum number of PPPoE logical interfaces that can be activated on the underlying interface.	**detail extensive** none
Duplicate Protection	State of PPPoE duplicate protection: **On** or **Off**. When duplicate protection is configured for the underlying interface, a dynamic PPPoE logical interface cannot be activated when an existing active logical interface is present for the same PPPoE client.	**detail extensive** none
Direct Connect	State of the configuration to ignore DSL Forum VSAs: **On** or **Off**. When configured, the router ignores any of these VSAs received from a directly connected CPE device on the interface.	**detail extensive** none
AC Name	Name of the access concentrator.	**detail extensive** none
Maximum labels	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	**detail extensive** none
Traffic statistics	Number and rate of bytes and packets received and transmitted on the specified interface set. • **Input bytes, Output bytes**—Number of bytes received and transmitted on the interface set. The value in this field also includes the Layer 2 overhead bytes for ingress or egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Input packets, Output packets**—Number of packets received and transmitted on the interface set.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled.	**extensive**
Local statistics	Number and rate of bytes and packets destined to the router.	**extensive**
Transit statistics	Number and rate of bytes and packets transiting the switch. NOTE: For Gigabit Ethernet intelligent queuing 2 (IQ2) interfaces, the logical interface egress statistics might not accurately reflect the traffic on the wire when output shaping is applied. Traffic management output shaping might drop packets after they are tallied by the **Output bytes** and **Output packets** interface counters. However, correct values display for both of these egress statistics when per-unit scheduling is enabled for the Gigabit Ethernet IQ2 physical interface, or when a single logical interface is actively using a shared scheduler.	**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
Flags	Information about protocol family flags. Possible values are described in the "Family Flags" section under *Common Output Fields Description*.	**detail extensive**

Table 146: show interfaces (Gigabit Ethernet) Output Fields (continued)

Field Name	Field Description	Level of Output
Donor interface	(Unnumbered Ethernet) Interface from which an unnumbered Ethernet interface borrows an IPv4 address.	**detail extensive** none
Preferred source address	(Unnumbered Ethernet) Secondary IPv4 address of the donor loopback interface that acts as the preferred source address for the unnumbered Ethernet interface.	**detail extensive** none
Input Filters	Names of any input filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parentheses next to all interfaces.	**detail extensive**
Output Filters	Names of any output filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parentheses next to all interfaces.	**detail extensive**
Mac-Validate Failures	Number of MAC address validation failures for packets and bytes. This field is displayed when MAC address validation is enabled for the logical interface.	**detail extensive** none
Addresses, Flags	Information about the address flags. Possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**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. Possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**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 interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**

The following table describes the output fields for the **show interfaces** (10–Gigabit Ethernet) command.

Field Name	Field Description	Level of Output
Physical interface	Name of the physical interface.	All levels
Enabled	State of the interface. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.	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
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
WAN-PHY mode	10-Gigabit Ethernet interface operating in Wide Area Network Physical Layer Device (WAN PHY) mode. WAN PHY allows 10-Gigabit Ethernet wide area links to use fiber-optic cables and other devices intended for SONET/SDH.	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**.	All levels
Auto-negotiation	(Gigabit Ethernet interfaces) Autonegotiation status: **Enabled** or **Disabled**.	All levels
Remote-fault	(Gigabit Ethernet interfaces) 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. Possible values are described in the “Device Flags” section under *Common Output Fields Description*.	All levels
Interface flags	Information about the interface. Possible values are described in the “Interface Flags” section under *Common Output Fields Description*.	All levels
Link flags	Information about the link. Possible values are described in the “Links Flags” section under *Common Output Fields Description*.	All levels
Wavelength	(10-Gigabit Ethernet dense wavelength-division multiplexing [DWDM] interfaces) Displays the configured wavelength, in nanometers (nm).	All levels
Frequency	(10-Gigabit Ethernet DWDM interfaces only) Displays the frequency associated with the configured wavelength, in terahertz (THz).	All levels
CoS queues	Number of CoS queues configured.	**detail extensive** none

Schedulers	(Gigabit Ethernet intelligent queuing 2 (IQ2) interfaces only) 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: 2002-04-26 10:52:40 PDT (04:33:20 ago).**	**detail extensive** none
Input Rate	Input rate in bits per second (bps) and packets per second (pps). The value in this field also includes the Layer 2 overhead bytes for ingress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.	None specified
Output Rate	Output rate in bps and pps. The value in this field also includes the Layer 2 overhead bytes for egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level.	None specified
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**
Egress account overhead	Layer 2 overhead in bytes that is accounted in the interface statistics for egress traffic.	**detail extensive**
Ingress account overhead	Layer 2 overhead in bytes that is accounted in the interface statistics for ingress traffic.	**detail extensive**
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. The value in this field also includes the Layer 2 overhead bytes for ingress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Output bytes**—Number of bytes transmitted on the interface. The value in this field also includes the Layer 2 overhead bytes for egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	**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 the Junos OS does not handle. • **L3 incompletes**—Number of incoming packets discarded because they failed Layer 3 (usually IPv4) 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 by configuring 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**
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. NOTE: In DPCs that are not of the enhanced type, such as DPC 40x 1GE R, DPCE 20x 1GE + 2x 10GE R, or DPCE 40x 1GE R, you might notice a discrepancy in the output of the **show interfaces** command because incoming packets might be counted in the Egress queues section of the output. This problem occurs on non-enhanced DPCs because the egress queue statistics are polled from IMQ (Inbound Message Queuing) block of the I-chip. The IMQ block does not differentiate between ingress and egress WAN traffic; as a result, the combined statistics are displayed in the egress queue counters on the Routing Engine. In a simple VPLS scenorio, if there is no MAC entry in DMAC table (by sending unidirectional traffic), traffic is flooded and the input traffic is accounted in IMQ. For bidirectional traffic (MAC entry in DMAC table), if the outgoing interface is on the same I-chip then both ingress and egress statistics are counted in a combined way. If the outgoing interface is on a different I-chip or FPC, then only egress statistics are accounted in IMQ. This behavior is expected with non-enhanced DPCs	**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**
Ingress queues	Total number of ingress queues supported on the specified interface. Displayed on IQ2 interfaces.	**extensive**
Queue counters (Ingress)	CoS queue number and its associated user-configured forwarding class name. Displayed on IQ2 interfaces. • **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**
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 routing device configuration, an alarm can ring the red or yellow alarm bell on the routing device, 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
OTN alarms	Active OTN alarms identified on the interface.	**detail extensive**
OTN defects	OTN defects received on the interface.	**detail extensive**
OTN FEC Mode	The FECmode configured on the interface. • **efec**—Enhanced forward error correction (EFEC) is configured to defect and correct bit errors. • **gfec**—G.709 Forward error correction (GFEC) mode is configured to detect and correct bit errors. • **none**—FEC mode is not configured.	**detail extensive**

OTN Rate	OTN mode. • **fixed-stuff-bytes**—Fixed stuff bytes 11.0957 Gbps. • **no-fixed-stuff-bytes**—No fixed stuff bytes 11.0491 Gbps. • **pass-through**—Enable OTN passthrough mode. • **no-pass-through**—Do not enable OTN passthrough mode.	**detail extensive**
OTN Line Loopback	Status of the line loopback, if configured for the DWDM OTN PIC. Its value can be: **enabled** or **disabled**.	**detail extensive**
OTN FEC statistics	The forward error correction (FEC) counters for the DWDM OTN PIC. • **Corrected Errors**—The count of corrected errors in the last second. • **Corrected Error Ratio**—The corrected error ratio in the last 25 seconds. For example, 1e-7 is 1 error per 10 million bits.	**detail extensive**
OTN FEC alarms	OTN FEC excessive or degraded error alarms triggered on the interface. • **FEC Degrade**—OTU FEC Degrade defect. • **FEC Excessive**—OTU FEC Excessive Error defect.	**detail extensive**
OTN OC	OTN OC defects triggered on the interface. • **LOS**—OC Loss of Signal defect. • **LOF**—OC Loss of Frame defect. • **LOM**—OC Loss of Multiframe defect. • **Wavelength Lock**—OC Wavelength Lock defect.	**detail extensive**
OTN OTU	OTN OTU defects detected on the interface • **AIS**—OTN AIS alarm. • **BDI**—OTN OTU BDI alarm. • **IAE**—OTN OTU IAE alarm. • **TTIM**—OTN OTU TTIM alarm. • **SF**—OTN ODU bit error rate fault alarm. • **SD**—OTN ODU bit error rate defect alarm. • **TCA-ES**—OTN ODU ES threshold alarm. • **TCA-SES**—OTN ODU SES threshold alarm. • **TCA-UAS**—OTN ODU UAS threshold alarm. • **TCA-BBE**—OTN ODU BBE threshold alarm. • **BIP**—OTN ODU BIP threshold alarm. • **BBE**—OTN OTU BBE threshold alarm. • **ES**—OTN OTU ES threshold alarm. • **SES**—OTN OTU SES threshold alarm. • **UAS**—OTN OTU UAS threshold alarm.	**detail extensive**
Received DAPI	Destination Access Port Interface (DAPI) from which the packets were received.	**detail extensive**
Received SAPI	Source Access Port Interface (SAPI) from which the packets were received.	**detail extensive**
Transmitted DAPI	Destination Access Port Interface (DAPI) to which the packets were transmitted.	**detail extensive**

Transmitted SAPI	Source Access Port Interface (SAPI) to which the packets were transmitted.	**detail extensive**
PCS statistics	(10-Gigabit Ethernet interfaces) Displays Physical Coding Sublayer (PCS) fault conditions from the WAN PHY or the LAN PHY device. • **Bit errors**—The number of seconds during which at least one bit error rate (BER) occurred while the PCS receiver is operating in normal mode. • **Errored blocks**—The number of seconds when at least one errored block occurred while the PCS receiver is operating in normal mode.	**detail extensive**
MAC statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC subsystem, including the following: • **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 or a cable 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 frames that exceed 1518 octets. • **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. • **Code violations**—Number of times an event caused the PHY to indicate “Data reception error” or “invalid data symbol error.”	**extensive**
OTN Received Overhead Bytes	APS/PCC0: 0x02, APS/PCC1: 0x11, APS/PCC2: 0x47, APS/PCC3: 0x58 Payload Type: 0x08	**extensive**
OTN Transmitted Overhead Bytes	APS/PCC0: 0x00, APS/PCC1: 0x00, APS/PCC2: 0x00, APS/PCC3: 0x00 Payload Type: 0x08	**extensive**

Filter statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC address filter subsystem. The filtering is done by the content-addressable memory (CAM) on the PIC. The filter examines a packet's source and destination MAC addresses to determine whether the packet should enter the system or be rejected. • **Input packet count**—Number of packets received from the MAC hardware that the filter processed. • **Input packet rejects**—Number of packets that the filter rejected because of either the source MAC address or the destination MAC address. • **Input DA rejects**—Number of packets that the filter rejected because the destination MAC address of the packet is not on the accept list. It is normal for this value to increment. When it increments very quickly and no traffic is entering the routing device from the far-end system, either there is a bad ARP entry on the far-end system, or multicast routing is not on and the far-end system is sending many multicast packets to the local routing device (which the routing device is rejecting). • **Input SA rejects**—Number of packets that the filter rejected because the source MAC address of the packet is not on the accept list. The value in this field should increment only if source MAC address filtering has been enabled. If filtering is enabled, if the value increments quickly, and if the system is not receiving traffic that it should from the far-end system, it means that the user-configured source MAC addresses for this interface are incorrect. • **Output packet count**—Number of packets that the filter has given to the MAC hardware. • **Output packet pad count**—Number of packets the filter padded to the minimum Ethernet size (60 bytes) before giving the packet to the MAC hardware. Usually, padding is done only on small ARP packets, but some very small IP packets can also require padding. If this value increments rapidly, either the system is trying to find an ARP entry for a far-end system that does not exist or it is misconfigured. • **Output packet error count**—Number of packets with an indicated error that the filter was given to transmit. These packets are usually aged packets or are the result of a bandwidth problem on the FPC hardware. On a normal system, the value of this field should not increment. • **CAM destination filters, CAM source filters**—Number of entries in the CAM dedicated to destination and source MAC address filters. There can only be up to 64 source entries. If source filtering is disabled, which is the default, the values for these fields should be 0.	**extensive**
PMA PHY	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET error information: • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem.	**extensive**

WIS section	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET error information: • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem. Subfields are: • **BIP-B1**—Bit interleaved parity for SONET section overhead • **SEF**—Severely errored framing • **LOL**—Loss of light • **LOF**—Loss of frame • **ES-S**—Errored seconds (section) • S**ES-S**—Severely errored seconds (section) • **SEFS-S**—Severely errored framing seconds (section)	**extensive**
WIS line	(10-Gigabit Ethernet interfaces, WAN PHY mode) Active alarms and defects, plus counts of specific SONET errors with detailed information. • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. State other than **OK** indicates a problem. Subfields are: • **BIP-B2**—Bit interleaved parity for SONET line overhead • **REI-L**—Remote error indication (near-end line) • **RDI-L**—Remote defect indication (near-end line) • **AIS-L**—Alarm indication signal (near-end line) • **BERR-SF**—Bit error rate fault (signal failure) • **BERR-SD**—Bit error rate defect (signal degradation) • **ES-L**—Errored seconds (near-end line) • **SES-L**—Severely errored seconds (near-end line) • **UAS-L**—Unavailable seconds (near-end line) • **ES-LFE**—Errored seconds (far-end line) • **SES-LFE**—Severely errored seconds (far-end line) • **UAS-LFE**—Unavailable seconds (far-end line)	**extensive**

WIS path	(10-Gigabit Ethernet interfaces, WAN PHY mode) Active alarms and defects, plus counts of specific SONET errors with detailed information. • **Seconds**—Number of seconds the defect has been active. • **Count**—Number of times that the defect has gone from inactive to active. • **State**—State of the error. Any state other than **OK** indicates a problem. Subfields are: • **BIP-B3**—Bit interleaved parity for SONET section overhead • **REI-P**—Remote error indication • **LOP-P**—Loss of pointer (path) • **AIS-P**—Path alarm indication signal • **RDI-P**—Path remote defect indication • **UNEQ-P**—Path unequipped • **PLM-P**—Path payload label mismatch • **ES-P**—Errored seconds (near-end STS path) • **SES-P**—Severely errored seconds (near-end STS path) • **UAS-P**—Unavailable seconds (near-end STS path) • **SES-PFE**—Severely errored seconds (far-end STS path) • **UAS-PFE**—Unavailable seconds (far-end STS path)	**extensive**

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 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**—Information from the link partner:
	  ◦ **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).
	  ◦ **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

Received path trace, Transmitted path trace	(10-Gigabit Ethernet interfaces, WAN PHY mode) SONET/SDH interfaces allow path trace bytes to be sent inband across the SONET/SDH link. Juniper Networks and other router manufacturers use these bytes to help diagnose misconfigurations and network errors by setting the transmitted path trace message so that it contains the system hostname and name of the physical interface. The received path trace value is the message received from the routing device at the other end of the fiber. The transmitted path trace value is the message that this routing device transmits.	**extensive**
Packet Forwarding Engine configuration	Information about the configuration of the Packet Forwarding Engine: • **Destination slot**—FPC slot number.	**extensive**
CoS information	Information about the CoS queue for the physical interface. • **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. Possible values are described in the “Logical Interface Flags” section under *Common Output Fields Description*.	All levels

VLAN-Tag	Rewrite profile applied to incoming or outgoing frames on the outer (**Out**) VLAN tag or for both the outer and inner (**In**) VLAN tags. • **push**—An outer VLAN tag is pushed in front of the existing VLAN tag. • **pop**—The outer VLAN tag of the incoming frame is removed. • **swap**—The outer VLAN tag of the incoming frame is overwritten with the user specified VLAN tag information. • **push**—An outer VLAN tag is pushed in front of the existing VLAN tag. • **push-push**—Two VLAN tags are pushed in from the incoming frame. • **swap-push**—The outer VLAN tag of the incoming frame is replaced by a user-specified VLAN tag value. A user-specified outer VLAN tag is pushed in front. The outer tag becomes an inner tag in the final frame. • **swap-swap**—Both the inner and the outer VLAN tags of the incoming frame are replaced by the user specified VLAN tag value. • **pop-swap**—The outer VLAN tag of the incoming frame is removed, and the inner VLAN tag of the incoming frame is replaced by the user-specified VLAN tag value. The inner tag becomes the outer tag in the final frame. • **pop-pop**—Both the outer and inner VLAN tags of the incoming frame are removed.	**brief detail extensive** none
Demux:	IP demultiplexing (demux) value that appears if this interface is used as the demux underlying interface. The output is one of the following: • Source Family Inet • Destination Family Inet	**detail extensive** none
Encapsulation	Encapsulation on the logical interface.	All levels
Protocol	Protocol family. Possible values are described in the "Protocol Field" section under *Common Output Fields Description*.	**detail extensive** none
MTU	Maximum transmission unit size on the logical interface.	**detail extensive** none
Maximum labels	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	**detail extensive** none
Traffic statistics	Number and rate of bytes and packets received and transmitted on the specified interface set. • **Input bytes, Output bytes**—Number of bytes received and transmitted on the interface set. The value in this field also includes the Layer 2 overhead bytes for ingress or egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Input packets, Output packets**—Number of packets received and transmitted on the interface set.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled.	**extensive**
Local statistics	Number and rate of bytes and packets destined to the routing device.	**extensive**

Transit statistics	Number and rate of bytes and packets transiting the switch. NOTE: For Gigabit Ethernet intelligent queuing 2 (IQ2) interfaces, the logical interface egress statistics might not accurately reflect the traffic on the wire when output shaping is applied. Traffic management output shaping might drop packets after they are tallied by the **Output bytes** and **Output packets** interface counters. However, correct values display for both of these egress statistics when per-unit scheduling is enabled for the Gigabit Ethernet IQ2 physical interface, or when a single logical interface is actively using a shared scheduler.	**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
Flags	Information about protocol family flags. Possible values are described in the "Family Flags" section under *Common Output Fields Description*.	**detail extensive**
Donor interface	(Unnumbered Ethernet) Interface from which an unnumbered Ethernet interface borrows an IPv4 address.	**detail extensive** none
Preferred source address	(Unnumbered Ethernet) Secondary IPv4 address of the donor loopback interface that acts as the preferred source address for the unnumbered Ethernet interface.	**detail extensive** none
Input Filters	Names of any input filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parenthesis next to all interfaces.	**detail extensive**
Output Filters	Names of any output filters applied to this interface. If you specify a precedence value for any filter in a dynamic profile, filter precedence values appear in parenthesis next to all interfaces.	**detail extensive**
Mac-Validate Failures	Number of MAC address validation failures for packets and bytes. This field is displayed when MAC address validation is enabled for the logical interface.	**detail extensive** none
Addresses, Flags	Information about the address flags. Possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**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 address flag (possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**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**

For Gigabit Ethernet IQ PICs, traffic and MAC statistics output varies. The following table describes the traffic and MAC statistics for two sample interfaces, each of which is sending traffic in packets of 500 bytes (including 478 bytes for the Layer 3 packet, 18 bytes for the Layer 2 VLAN traffic header, and 4 bytes for cyclic redundancy check [CRC] information). The **ge-0/3/0** interface is the inbound physical interface, and the **ge-0/0/0** interface is the outbound physical interface. On both interfaces, traffic is carried on logical unit **.50** (VLAN 50).

Table 147: Gigabit and 10 Gigabit Ethernet IQ PIC Traffic and MAC Statistics by Interface Type

Interface Type	Sample Command	Byte and Octet Counts Include	Comments
Inbound physical interface	**show interfaces ge-0/3/0 extensive**	Traffic statistics: Input bytes: 496 bytes per packet, representing the Layer 2 packet MAC statistics: Received octets: 500 bytes per packet, representing the Layer 2 packet + 4 bytes	The additional 4 bytes are for the CRC.
Inbound logical interface	**show interfaces ge-0/3/0.50 extensive**	Traffic statistics: Input bytes: 478 bytes per packet, representing the Layer 3 packet	
Outbound physical interface	**show interfaces ge-0/0/0 extensive**	Traffic statistics: Input bytes: 490 bytes per packet, representing the Layer 3 packet + 12 bytes MAC statistics: Received octets: 478 bytes per packet, representing the Layer 3 packet	For input bytes, the additional 12 bytes include 6 bytes for the destination MAC address plus 4 bytes for VLAN plus 2 bytes for the Ethernet type.
Outbound logical interface	**show interfaces ge-0/0/0.50 extensive**	Traffic statistics: Input bytes: 478 bytes per packet, representing the Layer 3 packet	

Table 148 on page 1299 lists the output fields for the **show interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 148: show interfaces 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
Link-level type	Encapsulation being used on the physical interface.	All levels
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
MTU	Maximum transmission unit size on the physical interface.	All levels
Link mode	Link mode: Full-duplex or Half-duplex.	
Speed	Speed at which the interface is running.	All levels
BPDU error	Bridge protocol data unit (BPDU) error: Detected or None	
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
Flow control	Flow control status: **Enabled** or **Disabled**.	All levels
Auto-negotiation	(Gigabit Ethernet interfaces) Autonegotiation status: **Enabled** or **Disabled**.	All levels
Remote-fault	(Gigabit Ethernet interfaces) 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 physical link.	All levels
CoS queues	Number of CoS queues configured.	**detail extensive** none
Current address	Configured MAC address.	**detail extensive** none

Table 148: show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
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: 2002-04-26 10:52:40 PDT (04:33:20 ago).**	**detail extensive** none
Input Rate	Input rate in bits per second (bps) and packets per second (pps).	None
Output Rate	Output rate in bps and pps.	None
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. 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
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**
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.	**detail extensive**

Table 148: show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
Input errors	Input errors on the interface. • **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 (usually IPv4) 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 by configuring the **ignore-l3-incompletes** . • **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**
Output errors	Output errors on the interface. • **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; therefore, for Gigabit Ethernet PICs, this number must 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 must 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 interfaces. • **MTU errors**—Number of packets whose size exceeded the MTU of the interface. • **Resource errors**—Sum of transmit drops.	**extensive**

Table 148: show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
Ingress queues	Total number of ingress queues supported on the specified interface.	**extensive**
Queue counters and queue number	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**
MAC statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC subsystem, including the following: • **Total octets** and **total packets**—Total number of octets and packets. • **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 or a cable 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**—There are two possible conditions regarding the number of oversized frames: ◦ Packet length exceeds 1518 octets, or ◦ Packet length exceeds MRU • **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. • **Code violations**—Number of times an event caused the PHY to indicate “Data reception error” or “invalid data symbol error.”	**extensive**

Table 148: show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
Filter statistics	**Receive** and **Transmit** statistics reported by the PIC's MAC address filter subsystem. The filtering is done by the content-addressable memory (CAM) on the PIC. The filter examines a packet's source and destination MAC addresses to determine whether the packet should enter the system or be rejected. • **Input packet count**—Number of packets received from the MAC hardware that the filter processed. • **Input packet rejects**—Number of packets that the filter rejected because of either the source MAC address or the destination MAC address. • **Input DA rejects**—Number of packets that the filter rejected because the destination MAC address of the packet is not on the accept list. It is normal for this value to increment. When it increments very quickly and no traffic is entering the device from the far-end system, either there is a bad ARP entry on the far-end system, or multicast routing is not on and the far-end system is sending many multicast packets to the local device (which the router is rejecting). • **Input SA rejects**—Number of packets that the filter rejected because the source MAC address of the packet is not on the accept list. The value in this field should increment only if source MAC address filtering has been enabled. If filtering is enabled, if the value increments quickly, and if the system is not receiving traffic that it should from the far-end system, it means that the user-configured source MAC addresses for this interface are incorrect. • **Output packet count**—Number of packets that the filter has given to the MAC hardware. • **Output packet pad count**—Number of packets the filter padded to the minimum Ethernet size (60 bytes) before giving the packet to the MAC hardware. Usually, padding is done only on small ARP packets, but some very small IP packets can also require padding. If this value increments rapidly, either the system is trying to find an ARP entry for a far-end system that does not exist or it is misconfigured. • **Output packet error count**—Number of packets with an indicated error that the filter was given to transmit. These packets are usually aged packets or are the result of a bandwidth problem on the FPC hardware. On a normal system, the value of this field should not increment. • **CAM destination filters, CAM source filters**—Number of entries in the CAM dedicated to destination and source MAC address filters. There can only be up to 64 source entries. If source filtering is disabled, which is the default, the values for these fields must be 0.	**extensive**
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.	**extensive**
Packet Forwarding Engine configuration	Information about the configuration of the Packet Forwarding Engine: • **Destination slot**—FPC slot number.	**extensive**

Table 148: show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
CoS information	Information about the CoS queue for the physical interface. • **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**
Interface transmit statistics	Status of the **interface-transmit-statistics** configuration: Enabled or Disabled.	**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**
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
Traffic statistics	Number and rate of bytes and packets received and transmitted on the specified interface set. • **Input bytes, Output bytes**—Number of bytes received and transmitted on the interface set. The value in this field also includes the Layer 2 overhead bytes for ingress or egress traffic on Ethernet interfaces if you enable accounting of Layer 2 overhead at the PIC level or the logical interface level. • **Input packets, Output packets**—Number of packets received and transmitted on the interface set.	**detail extensive**

Table 148: show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
Local statistics	Number and rate of bytes and packets destined to the device.	**extensive**
Transit statistics	Number and rate of bytes and packets transiting the switch. NOTE: For Gigabit Ethernet intelligent queuing 2 (IQ2) interfaces, the logical interface egress statistics might not accurately reflect the traffic on the wire when output shaping is applied. Traffic management output shaping might drop packets after they are tallied by the **Output bytes** and **Output packets** interface counters. However, correct values display for both of these egress statistics when per-unit scheduling is enabled for the Gigabit Ethernet IQ2 physical interface, or when a single logical interface is actively using a shared scheduler.	**extensive**
Security	Security zones that interface belongs to.	**extensive**
Flow Input statistics	Statistics on packets received by flow module.	**extensive**
Flow Output statistics	Statistics on packets sent by flow module.	**extensive**
Flow error statistics (Packets dropped due to)	Statistics on errors in the flow module.	**extensive**
Protocol	Protocol family.	**detail extensive** none
MTU	Maximum transmission unit size on the logical interface.	**detail extensive** none
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
Flags	Information about protocol family flags. .	**detail extensive**
Addresses, Flags	Information about the address flags..	**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 interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**

Sample Output Gigabit Ethernet

show interfaces (Gigabit Ethernet)

```
user@host> show interfaces ge-3/0/2
Physical interface: ge-3/0/2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 35
  Link-level type: 52, MTU: 1522, Speed: 1000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  CoS queues     : 4 supported, 4 maximum usable queues
  Current address: 00:00:5e:00:53:7c, Hardware address: 00:00:5e:00:53:7c
  Last flapped   : 2006-08-10 17:25:10 PDT (00:01:08 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Ingress rate at Packet Forwarding Engine      : 0 bps (0 pps)
  Ingress drop rate at Packet Forwarding Engine : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None

  Logical interface ge-3/0/2.0 (Index 72) (SNMP ifIndex 69)
    Flags: SNMP-Traps 0x4000
    VLAN-Tag [ 0x8100.512 0x8100.513 ] In(pop-swap 0x8100.530) Out(swap-push
    0x8100.512 0x8100.513)
    Encapsulation: VLAN-CCC
    Egress account overhead: 100
    Ingress account overhead: 90
    Input packets : 0
    Output packets: 0
    Protocol ccc, MTU: 1522
      Flags: Is-Primary
```

show interfaces (Gigabit Ethernet on MX Series Routers)

```
user@host> show interfaces ge-2/2/2
Physical interface: ge-2/2/2, Enabled, Physical link is Up
  Interface index: 156, SNMP ifIndex: 188
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, MAC-REWRITE Error: None,
 Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 4 maximum usable queues
  Schedulers     : 0
  Current address: 00:00:5e:00:53:c0, Hardware address: 00:00:5e:00:53:76
  Last flapped   : 2008-09-05 16:44:30 PDT (3d 01:04 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : None
  Active defects : None
  Logical interface ge-2/2/2.0 (Index 82) (SNMP ifIndex 219)
        Flags: Up SNMP-Traps 0x4004000 Encapsulation: ENET2
    Input packets : 10232
    Output packets: 10294
    Protocol inet, MTU: 1500
      Flags: Sendbcast-pkt-to-re
```

```
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 203.0.113/24, Local: 203.0.113.1, Broadcast: 203.0.113.255
    Protocol inet6, MTU: 1500
     Max nh cache: 4, New hold nh limit: 100000, Curr nh cnt: 4, Curr new hold
cnt: 4, NH drop cnt: 0
      Flags: Is-Primary
      Addresses, Flags: Is-Default Is-Preferred Is-Primary
        Destination: 2001:db8:/32, Local: 2001:db8::5
      Addresses, Flags: Is-Preferred
        Destination: 2001:db8:1::/32, Local: 2001:db8:223:9cff:fe9f:3e78
    Protocol multiservice, MTU: Unlimited
      Flags: Is-Primary
```

show interfaces (link degrade status)

```
user@host> show interfaces et-3/0/0
Physical interface: et-3/0/0, Enabled, Physical link is Down
  Interface index: 157, SNMP ifIndex: 537
  Link-level type: Ethernet, MTU: 1514, MRU: 0, Speed: 100Gbps, BPDU Error: None,
 Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 54:e0:32:23:9d:38, Hardware address: 54:e0:32:23:9d:38
  Last flapped   : 2014-06-18 02:36:38 PDT (02:50:50 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : LINK
  Active defects : LINK
  PCS statistics                      Seconds
    Bit errors                             0
    Errored blocks                         0
  Link Degrade* :
  Link Monitoring                   :  Enable
  Link Degrade Set Threshold:       :  1E-7
  Link Degrade Clear Threshold:     :  1E-12
  Estimated BER                     :  1E-7
  Link-degrade event                :  Seconds     Count    State
                                           782         1     Defect Active
```

show interfaces extensive (Gigabit Ethernet on MX Series Routers showing interface transmit statistics configuration)

```
user@host> show interfaces ge-2/1/2 extensive | match "output|interface"
Physical interface: ge-2/1/2, Enabled, Physical link is Up
  Interface index: 151, SNMP ifIndex: 530, Generation: 154
  Interface flags: SNMP-Traps Internal: 0x4000
   Output bytes  :         240614363944            772721536 bps
   Output packets:           3538446506              1420444 pps
    Direction : Output
  Interface transmit statistics: Enabled

  Logical interface ge-2/1/2.0 (Index 331) (SNMP ifIndex 955) (Generation 146)
     Output bytes  :         195560312716            522726272 bps
     Output packets:           4251311146              1420451 pps

user@host> show interfaces ge-5/2/0.0 statistics detail
Logical interface ge-5/2/0.0 (Index 71) (SNMP ifIndex 573) (Generation 135)
    Flags: SNMP-Traps 0x4000 Encapsulation: ENET2
```

```
  Egress account overhead: 100
  Ingress account overhead: 90
  Traffic statistics:
   Input  bytes  :               271524
   Output bytes  :             37769598
   Input  packets:                 3664
   Output packets:               885790
   IPv6 transit statistics:
    Input  bytes  :                    0
    Output bytes  :             16681118
    Input  packets:                    0
    Output packets:               362633
  Local statistics:
   Input  bytes  :               271524
   Output bytes  :               308560
   Input  packets:                 3664
   Output packets:                 3659
  Transit statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :             37461038                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:               882131                    0 pps
   IPv6 transit statistics:
    Input  bytes  :                    0     0 bps
    Output bytes  :             16681118     0 bps
    Input  packets:                    0     0 pps
    Output packets:               362633     0 pps
```

show interfaces brief (Gigabit Ethernet)

```
user@host> show interfaces ge-3/0/2 brief
Physical interface: ge-3/0/2, Enabled, Physical link is Up
  Link-level type: 52, MTU: 1522, Speed: 1000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface ge-3/0/2.0
    Flags: SNMP-Traps 0x4000
    VLAN-Tag [ 0x8100.512 0x8100.513 ] In(pop-swap 0x8100.530) Out(swap-push
    0x8100.512 0x8100.513)
    Encapsulation: VLAN-CCC
    ccc

  Logical interface ge-3/0/2.32767
    Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x0000.0 ]  Encapsulation: ENET2
```

show interfaces detail (Gigabit Ethernet)

```
user@host> show interfaces ge-3/0/2 detail
Physical interface: ge-3/0/2, Enabled, Physical link is Up
  Interface index: 167, SNMP ifIndex: 35, Generation: 177
  Link-level type: 52, MTU: 1522, Speed: 1000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
```

```
  CoS queues     : 4 supported, 4 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:7c, Hardware address: 00:00:5e:00:53:7c
  Last flapped   : 2006-08-09 17:17:00 PDT (01:31:33 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
  Ingress traffic statistics at Packet Forwarding Engine:
   Input  bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Drop   bytes  :                    0                    0 bps
   Drop   packets:                    0                    0 pps
  Ingress queues: 4 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    0                    0                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                   0                    0                    0

  Egress queues: 4 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    0                    0                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                   0                    0                    0

  Active alarms  : None
  Active defects : None

  Logical interface ge-3/0/2.0 (Index 72) (SNMP ifIndex 69) (Generation 140)
    Flags: SNMP-Traps 0x4000
    VLAN-Tag [0x8100.512 0x8100.513 ] In(pop-swap 0x8100.530)
Out(swap-push 0x8100.512 0x8100.513)
    Encapsulation: VLAN-CCC
    Egress account overhead: 100
    Ingress account overhead: 90
    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 ccc, MTU: 1522, Generation: 149, Route table: 0
     Flags: Is-Primary

  Logical interface ge-3/0/2.32767 (Index 71) (SNMP ifIndex 70)
  (Generation 139)
   Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x0000.0 ]  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
```

show interfaces extensive (Gigabit Ethernet IQ2)

```
user@host> show interfaces ge-7/1/3 extensive
Physical interface: ge-7/1/3, Enabled, Physical link is Up
  Interface index: 170, SNMP ifIndex: 70, Generation: 171
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4004000
  Link flags     : None
  CoS queues     : 8 supported, 4 maximum usable queues
  Schedulers     : 256
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:74, Hardware address: 00:00:5e:00:53:74
  Last flapped   : 2007-11-07 21:31:41 PST (02:03:33 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :          38910844056                 7952 bps
   Output bytes  :              7174605                 8464 bps
   Input  packets:            418398473                   11 pps
   Output packets:                78903                   12 pps
   IPv6 transit statistics:
    Input  bytes  :                   0
    Output bytes  :                   0
    Input  packets:                   0
    Output packets:                   0
  Ingress traffic statistics at Packet Forwarding Engine:
   Input  bytes  :          38910799145                 7952 bps
   Input  packets:            418397956                   11 pps
   Drop   bytes  :                   0                    0 bps
   Drop   packets:                   0                    0 pps
  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: 1, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0,
```

```
  FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Ingress queues: 4 supported, 4 in use
Queue counters:       Queued packets  Transmitted packets      Dropped packets

  0 best-effort            418390823            418390823                    0

  1 expedited-fo                   0                    0                    0

  2 assured-forw                   0                    0                    0

  3 network-cont                7133                 7133                    0

Egress queues: 4 supported, 4 in use
Queue counters:       Queued packets  Transmitted packets      Dropped packets

  0 best-effort                1031                 1031                    0

  1 expedited-fo                   0                    0                    0

  2 assured-forw                   0                    0                    0

  3 network-cont               77872                77872                    0

Active alarms  : None
Active defects : None
MAC statistics:                      Receive         Transmit
  Total octets                    38910844056          7174605
  Total packets                     418398473            78903
  Unicast packets                 408021893366            1026
  Broadcast packets                         10              12
  Multicast packets                 418398217            77865
  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  OTN Received Overhead Bytes:
  APS/PCC0: 0x02, APS/PCC1: 0x11, APS/PCC2: 0x47, APS/PCC3: 0x58
  Payload Type: 0x08
OTN Transmitted Overhead Bytes:
  APS/PCC0: 0x00, APS/PCC1: 0x00, APS/PCC2: 0x00, APS/PCC3: 0x00
  Payload Type: 0x08
Filter statistics:
  Input packet count               418398473
  Input packet rejects                479
  Input DA rejects                    479
  Input SA rejects                      0
  Output packet count                                  78903
  Output packet pad count                                  0
  Output packet error count                                0
  CAM destination filters: 0, CAM source filters: 0
Autonegotiation information:
  Negotiation status: Complete
  Link partner:
      Link mode: Full-duplex, Flow control: Symmetric/Asymmetric,
      Remote fault: OK
  Local resolution:
```

```
        Flow control: Symmetric, Remote fault: Link OK
  Packet Forwarding Engine configuration:
    Destination slot: 7
  CoS information:
    Direction : Output
    CoS transmit queue          Bandwidth          Buffer     Priority   Limit
                              %            bps     %           usec
    0 best-effort            95      950000000    95              0
low    none
    3 network-control         5       50000000     5              0
low    none
    Direction : Input
    CoS transmit queue          Bandwidth          Buffer     Priority   Limit
                              %            bps     %           usec
    0 best-effort            95      950000000    95              0
low    none
    3 network-control         5       50000000     5              0
low    none

  Logical interface ge-7/1/3.0 (Index 70) (SNMP ifIndex 85) (Generation 150)
    Flags: SNMP-Traps Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :              812400
     Output bytes  :             1349206
     Input  packets:                9429
     Output packets:                9449
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :                   0
      Input  packets:                   0
      Output packets:                   0
    Local statistics:
     Input  bytes  :              812400
     Output bytes  :             1349206
     Input  packets:                9429
     Output packets:                9449
    Transit statistics:
     Input  bytes  :                   0                 7440 bps
     Output bytes  :                   0                 7888 bps
     Input  packets:                   0                   10 pps
     Output packets:                   0                   11 pps
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :                   0
      Input  packets:                   0
      Output packets:                   0
    Protocol inet, MTU: 1500, Generation: 169, Route table: 0
      Flags: Is-Primary, Mac-Validate-Strict
      Mac-Validate Failures: Packets: 0, Bytes: 0
      Addresses, Flags: Is-Preferred Is-Primary
      Input Filters: F1-ge-3/0/1.0-in, F3-ge-3/0/1.0-in
      Output Filters: F2-ge-3/0/1.0-out (53)
      Destination: 203.0.113/24, Local: 203.0.113.2, Broadcast: 203.0.113.255,
        Generation: 196
    Protocol multiservice, MTU: Unlimited, Generation: 170, Route table: 0
      Flags: Is-Primary
      Policer: Input: __default_arp_policer__
```

NOTE: For Gigabit Ethernet intelligent queuing 2 (IQ2) interfaces, the logical interface egress statistics displayed in the **show interfaces** command output might not accurately reflect the traffic on the wire when output shaping is applied. Traffic management output

shaping might drop packets after they are tallied by the interface counters. For detailed information, see the description of the logical interface **Transit statistics** fields in Table 146 on page 1270.

show interfaces (Gigabit Ethernet Unnumbered Interface)

```
user@host> show interfaces ge-3/2/0
Physical interface: ge-3/2/0, Enabled, Physical link is Up
  Interface index: 148, SNMP ifIndex: 50
  Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 4 maximum usable queues
  Current address: 00:00:5e:00:53:f8, Hardware address: 00:00:5e:00:53:f8
  Last flapped   : 2006-10-27 04:42:23 PDT (08:01:52 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 624 bps (1 pps)
  Active alarms  : None
  Active defects : None

  Logical interface ge-3/2/0.0 (Index 67) (SNMP ifIndex 85)
    Flags: SNMP-Traps Encapsulation: ENET2
    Input packets : 0
    Output packets: 6
    Protocol inet, MTU: 1500
      Flags: Unnumbered
      Donor interface: lo0.0 (Index 64)
      Preferred source address: 203.0.113.22
```

show interfaces (ACI Interface Set Configured)

```
user@host> show interfaces ge-1/0/0.4001
  Logical interface ge-1/0/0.4001 (Index 340) (SNMP ifIndex 548)
    Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.4001 ]  Encapsulation: PPP-over-

    Ethernet
    ACI VLAN:
      Dynamic Profile: aci-vlan-set-profile
    PPPoE:
      Dynamic Profile: aci-vlan-pppoe-profile,
      Service Name Table: None,
      Max Sessions: 32000, Max Sessions VSA Ignore: Off,
      Duplicate Protection: On, Short Cycle Protection: Off,
      Direct Connect: Off,
      AC Name: nbc
    Input packets : 9
    Output packets: 8
    Protocol multiservice, MTU: Unlimited
```

show interfaces (ALI Interface Set)

```
user@host> show interfaces ge-1/0/0.10
  Logical interface ge-1/0/0.10 (Index 346) (SNMP ifIndex 554) (Generation 155)
    Flags: Up SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.10 ]  Encapsulation: ENET2
    Line Identity:
```

```
    Dynamic Profile: ali-set-profile
    Circuit-id Remote-id Accept-no-ids
  PPPoE:
    Dynamic Profile: ali-vlan-pppoe-profile,
    Service Name Table: None,
    Max Sessions: 32000, Max Sessions VSA Ignore: Off,
    Duplicate Protection: On, Short Cycle Protection: Off,
    Direct Connect: Off,
    AC Name: nbc
  Input packets : 9
  Output packets: 8
  Protocol multiservice, MTU: Unlimited
```

Sample Output Gigabit Ethernet

show interfaces extensive (10-Gigabit Ethernet, LAN PHY Mode, IQ2)

```
user@host> show interfaces xe-5/0/0 extensive
Physical interface: xe-5/0/0, Enabled, Physical link is Up
  Interface index: 177, SNMP ifIndex: 99, Generation: 178
  Link-level type: Ethernet, MTU: 1518, LAN-PHY mode, Speed: 10Gbps, Loopback:
None, Source filtering: Enabled,
  Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 4 maximum usable queues
  Schedulers     : 1024
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:f6, Hardware address: 00:00:5e:00:53:f6
  Last flapped   : Never
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :          6970332384                 0 bps
   Output bytes  :                   0                 0 bps
   Input  packets:            81050506                 0 pps
   Output packets:                   0                 0 pps
   IPv6 transit statistics:
    Input  bytes  :                  0
    Output bytes  :                  0
    Input  packets:                  0
    Output packets:                  0
  Ingress traffic statistics at Packet Forwarding Engine:
   Input  bytes  :          6970299398                 0 bps
   Input  packets:            81049992                 0 pps
   Drop   bytes  :                   0                 0 bps
   Drop   packets:                   0                 0 pps
  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
  Ingress queues: 4 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort             81049992             81049992                   0

    1 expedited-fo                   0                    0                   0
```

```
  2 assured-forw                  0                    0                    0

  3 network-cont                  0                    0                    0

Egress queues: 4 supported, 4 in use
Queue counters:       Queued packets  Transmitted packets      Dropped packets

  0 best-effort                   0                    0                    0

  1 expedited-fo                  0                    0                    0

  2 assured-forw                  0                    0                    0

  3 network-cont                  0                    0                    0

Active alarms  : None
Active defects : None
PCS statistics                      Seconds
  Bit errors                             0
  Errored blocks                         0
MAC statistics:                      Receive         Transmit
  Total octets                    6970332384                0
  Total packets                     81050506                0
  Unicast packets                   81050000                0
  Broadcast packets                       506               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
Filter statistics:
  Input packet count                81050506
  Input packet rejects                    506
  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: 0, CAM source filters: 0
Packet Forwarding Engine configuration:
  Destination slot: 5
CoS information:
  Direction : Output
  CoS transmit queue          Bandwidth              Buffer Priority   Limit
                          %           bps     %         usec
  0 best-effort          95     950000000    95         0          low     none
  3 network-control       5      50000000     5         0          low     none

  Direction : Input
  CoS transmit queue          Bandwidth              Buffer Priority   Limit
                          %           bps     %         usec
  0 best-effort          95     950000000    95         0          low     none
  3 network-control       5      50000000     5         0          low     none

Logical interface xe-5/0/0.0 (Index 71) (SNMP ifIndex 95) (Generation 195)
```

```
    Flags: SNMP-Traps 0x4000 VLAN-Tag [ 0x8100.100 ]  Encapsulation: ENET2
    Egress account overhead: 100
    Ingress account overhead: 90
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                  46
     Input  packets:                   0
     Output packets:                   1
     IPv6 transit statistics:
      Input  bytes  :                  0
      Output bytes  :                  0
      Input  packets:                  0
      Output packets:                  0
    Local statistics:
     Input  bytes  :                   0
     Output bytes  :                  46
     Input  packets:                   0
     Output packets:                   1
    Transit 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
    Protocol inet, MTU: 1500, Generation: 253, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 192.0.2/24, Local: 192.0.2.1, Broadcast: 192.0.2.255,
Generation: 265
    Protocol multiservice, MTU: Unlimited, Generation: 254, Route table: 0
      Flags: None
      Policer: Input: __default_arp_policer__
```

show interfaces extensive (10-Gigabit Ethernet, WAN PHY Mode)

```
user@host> show interfaces xe-1/0/0 extensive
Physical interface: xe-1/0/0, Enabled, Physical link is Up
  Interface index: 141, SNMP ifIndex: 34, Generation: 47
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Loopback: Disabled
  WAN-PHY mode
  Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps 16384
  Link flags     : None
  CoS queues     : 4 supported
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:9d, Hardware address: 00:00:5e:00:53:9d
  Last flapped   : 2005-07-07 11:22:34 PDT (3d 12:28 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0                    0 bps
   Output bytes  :                   0                    0 bps
   Input  packets:                   0                    0 pps
   Output packets:                   0                    0 pps
  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,
    HS Link CRC errors: 0, HS Link FIFO overflows: 0,
```

```
  Resource errors: 0
 Output errors:
  Carrier transitions: 1, Errors: 0, Drops: 0, Collisions: 0,
  Aged packets: 0, FIFO errors: 0, HS link CRC errors: 0, MTU errors: 0,
  Resource errors: 0
 Queue counters:       Queued packets  Transmitted packets     Dropped packets
  0 best-effort                    0                    0                   0
  1 expedited-fo                   0                    0                   0
  2 assured-forw                   0                    0                   0
  3 network-cont                   0                    0                   0
 Active alarms : LOL, LOS, LBL
 Active defects: LOL, LOS, LBL, SEF, AIS-L, AIS-P
 PCS statistics             Seconds       Count
  Bit errors                      0           0
  Errored blocks                  0           0
 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
 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: 0, CAM source filters: 0
 PMA PHY:              Seconds        Count  State
  PLL lock                   0            0  OK
  PHY light              63159            1  Light Missing
WIS section:
  BIP-B1                     0            0
  SEF                   434430       434438  Defect Active
  LOS                   434430            1  Defect Active
  LOF                   434430            1  Defect Active
  ES-S                  434430
  SES-S                 434430
  SEFS-S                434430
WIS line:
  BIP-B2                     0            0
  REI-L                      0            0
  RDI-L                      0            0  OK
  AIS-L                 434430            1  Defect Active
  BERR-SF                    0            0  OK
  BERR-SD                    0            0  OK
  ES-L                  434430
  SES-L                 434430
  UAS-L                 434420
  ES-LFE                     0
```

```
    SES-LFE                  0
    UAS-LFE                  0
  WIS path:
    BIP-B3                   0            0
    REI-P                    0            0
    LOP-P                    0            0  OK
    AIS-P               434430            1  Defect Active
    RDI-P                    0            0  OK
    UNEQ-P                   0            0  OK
    PLM-P                    0            0  OK
    ES-P                434430
    SES-P               434430
    UAS-P               434420
    ES-PFE                   0
    SES-PFE                  0
    UAS-PFE                  0
  Received path trace:
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
  Transmitted path trace: orissa so-1/0/0
    6f 72 69 73 73 61 20 73 6f 2d 31 2f 30 2f 30 00   orissa so-1/0/0.
    Packet Forwarding Engine configuration:
      Destination slot: 1
    CoS information:
      CoS transmit queue          Bandwidth          Buffer     Priority   Limit
                            %            bps     %    bytes
      0 best-effort        95      950000000    95        0          low     none
      3 network-control     5       50000000     5        0          low     none
```

show interfaces extensive (10-Gigabit Ethernet, DWDM OTN PIC)

```
user@host> show interfaces ge-7/0/0 extensive
Physical interface: ge-7/0/0, Enabled, Physical link is Down
  Interface index: 143, SNMP ifIndex: 508, Generation: 208
  Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, BPDU Error: None,
  MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled,
  Flow control: Enabled
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x4000
  Link flags     : None
  Wavelength     : 1550.12 nm, Frequency: 193.40 THz
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:72, Hardware address: 00:00:5e:00:53:72
  Last flapped   : 2011-04-20 15:48:54 PDT (18:39:49 ago)
  Statistics last cleared: Never
  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: 2, 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: 8 supported, 4 in use
Queue counters:       Queued packets  Transmitted packets      Dropped packets

  0 best-effort                    0                    0                    0

  1 expedited-fo                   0                    0                    0

  2 assured-forw                   0                    0                    0

  3 network-cont
Queue number:         Mapped forwarding classes
  0                   best-effort
  1                   expedited-forwarding
  2                   assured-forwarding
  3                   network-control
Active alarms  : LINK
Active defects : LINK
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
  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
OTN alarms            :   None
OTN defects           :   None
OTN FEC Mode       : GFEC
OTN Rate              : Fixed Stuff Bytes 11.0957Gbps
OTN Line Loopback : Enabled
OTN FEC statistics :
  Corrected Errors                                    0
  Corrected Error Ratio (          0 sec average)     0e-0
OTN FEC alarms:          Seconds        Count  State
  FEC Degrade                  0            0  OK
  FEC Excessive                0            0  OK
OTN OC:                  Seconds        Count  State
  LOS                          2            1  OK
  LOF                      67164            2  Defect Active
```

```
    LOM                       67164           71  Defect Active
    Wavelength Lock               0            0  OK
  OTN OTU:
    AIS                           0            0  OK
    BDI                       65919         4814  Defect Active
    IAE                       67158            1  Defect Active
    TTIM                          7            1  OK
    SF                        67164            2  Defect Active
    SD                        67164            3  Defect Active
    TCA-ES                        0            0  OK
    TCA-SES                       0            0  OK
    TCA-UAS                      80           40  OK
    TCA-BBE                       0            0  OK
    BIP                           0            0  OK
    BBE                           0            0  OK
    ES                            0            0  OK
    SES                           0            0  OK
    UAS                         587            0  OK
    Received DAPI:
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
    Received SAPI:
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
    Transmitted DAPI:
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
    Transmitted SAPI:
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
  OTN Received Overhead Bytes:
    APS/PCC0: 0x02, APS/PCC1: 0x42, APS/PCC2: 0xa2, APS/PCC3: 0x48
    Payload Type: 0x03
  OTN Transmitted Overhead Bytes:
    APS/PCC0: 0x00, APS/PCC1: 0x00, APS/PCC2: 0x00, APS/PCC3: 0x00
    Payload Type: 0x03
  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: 0, CAM source filters: 0
  Packet Forwarding Engine configuration:
    Destination slot: 7
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority
Limit
                              %            bps     %           usec
    0 best-effort            95     9500000000    95              0      low
none
    3 network-control         5      500000000     5              0      low
none
    ...
```

show interfaces extensive (10-Gigabit Ethernet, LAN PHY Mode, Unidirectional Mode)

```
user@host> show interfaces xe-7/0/0 extensive
Physical interface: xe-7/0/0, Enabled, Physical link is Up
  Interface index: 173, SNMP ifIndex: 212, Generation: 174
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps,
Unidirectional: Enabled,
```

```
  Loopback: None, Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
...
```

show interfaces extensive (10-Gigabit Ethernet, LAN PHY Mode, Unidirectional Mode, Transmit-Only)

```
user@host> show interfaces xe-7/0/0—tx extensive
Physical interface: xe-7/0/0-tx, Enabled, Physical link is Up
  Interface index: 176, SNMP ifIndex: 137, Generation: 177
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps,
Unidirectional: Tx-Only
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:83, Hardware address: 00:00:5e:00:53:83
  Last flapped   : 2007-06-01 09:08:19 PDT (3d 02:31 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                    0                   0 bps
   Output bytes  :      322891152287160          9627472888 bps
   Input  packets:                    0                   0 pps
   Output packets:         328809727380             1225492 pps

...

  Filter statistics:
    Output packet count           328810554250
    Output packet pad count                  0
    Output packet error count                0
...

  Logical interface xe-7/0/0-tx.0 (Index 73) (SNMP ifIndex 138) (Generation 139)

    Flags: SNMP-Traps Encapsulation: ENET2
    Egress account overhead: 100
    Ingress account overhead: 90
    Traffic statistics:
     Input  bytes  :                    0
     Output bytes  :      322891152287160
     Input  packets:                    0
     Output packets:         328809727380
     IPv6 transit 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  :      322891152287160          9627472888 bps
     Input  packets:                    0                   0 pps
     Output packets:         328809727380             1225492 pps
     IPv6 transit statistics:
      Input  bytes  :                   0
      Output bytes  :                   0
```

```
      Input  packets:                    0
      Output packets:                    0
    Protocol inet, MTU: 1500, Generation: 147, Route table: 0
      Addresses, Flags: Is-Preferred Is-Primary
        Destination: 10.11.12/24, Local: 10.11.12.13, Broadcast: 10.11.12.255,
Generation: 141
    Protocol multiservice, MTU: Unlimited, Generation: 148, Route table: 0
      Flags: None
      Policer: Input: __default_arp_policer__
```

show interfaces extensive (10-Gigabit Ethernet, LAN PHY Mode, Unidirectional Mode, Receive-Only)

```
user@host> show interfaces xe-7/0/0—rx extensive
Physical interface: xe-7/0/0-rx, Enabled, Physical link is Up
  Interface index: 174, SNMP ifIndex: 118, Generation: 175
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps,
Unidirectional: Rx-Only
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:83, Hardware address: 00:00:5e:00:53:83
  Last flapped   : 2007-06-01 09:08:22 PDT (3d 02:31 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :      322857456303482           9627496104 bps
   Output bytes  :                   0                    0 bps
   Input  packets:         328775413751              1225495 pps
   Output packets:                   0                    0 pps

...

  Filter statistics:
    Input packet count            328775015056
    Input packet rejects                     1
    Input DA rejects                         0

...

  Logical interface xe-7/0/0-rx.0 (Index 72) (SNMP ifIndex 120) (Generation 138)

    Flags: SNMP-Traps Encapsulation: ENET2
    Traffic statistics:
     Input  bytes  :      322857456303482
     Output bytes  :                   0
     Input  packets:         328775413751
     Output packets:                   0
     IPv6 transit 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  :      322857456303482           9627496104 bps
     Output bytes  :                   0                    0 bps
```

```
      Input  packets:         328775413751              1225495 pps
      Output packets:                    0                    0 pps
      IPv6 transit statistics:
       Input  bytes  :                   0
       Output bytes  :                   0
       Input  packets:                   0
       Output packets:                   0
     Protocol inet, MTU: 1500, Generation: 145, Route table: 0
       Addresses, Flags: Is-Preferred Is-Primary
         Destination: 192.0.2/24, Local: 192.0.2.1, Broadcast: 192.0.2.255,
Generation: 139
     Protocol multiservice, MTU: Unlimited, Generation: 146, Route table: 0
       Flags: None
       Policer: Input: __default_arp_policer__
```

Sample Output

Sample Output SRX Gigabit Ethernet

```
user@host> show interfaces ge-0/0/1
Physical interface: ge-0/0/1, Enabled, Physical link is Down
  Interface index: 135, SNMP ifIndex: 510
  Link-level type: Ethernet, MTU: 1514, Link-mode: Full-duplex, Speed: 1000mbps,

  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:01, Hardware address: 00:00:5e:00:53:01
  Last flapped   : 2015-05-12 08:36:59 UTC (1w1d 22:42 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : LINK
  Active defects : LINK
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/1.0 (Index 71) (SNMP ifIndex 514)
    Flags: Device-Down SNMP-Traps 0x0 Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Security: Zone: public
    Protocol inet, MTU: 1500
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255
```

Sample Output SRX Gigabit Ethernet

```
user@host> show interfaces ge-0/0/1
Physical interface: ge-0/0/1, Enabled, Physical link is Down
  Interface index: 135, SNMP ifIndex: 510
  Link-level type: Ethernet, MTU: 1514, Link-mode: Full-duplex, Speed: 1000mbps,

  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running Down
```

```
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Current address: 00:00:5e:00:53:01, Hardware address: 00:00:5e:00:53:01
  Last flapped   : 2015-05-12 08:36:59 UTC (1w1d 22:42 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Active alarms  : LINK
  Active defects : LINK
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/1.0 (Index 71) (SNMP ifIndex 514)
    Flags: Device-Down SNMP-Traps 0x0 Encapsulation: ENET2
    Input packets : 0
    Output packets: 0
    Security: Zone: public
    Protocol inet, MTU: 1500
      Flags: Sendbcast-pkt-to-re
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255
```

show interfaces detail (Gigabit Ethernet)

```
user@host> show interfaces ge-0/0/1 detail
Physical interface: ge-0/0/1, Enabled, Physical link is Down
  Interface index: 135, SNMP ifIndex: 510, Generation: 138
  Link-level type: Ethernet, MTU: 1514, Link-mode: Full-duplex, Speed: 1000mbps,
 BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering:
 Disabled,
  Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:01, Hardware address: 00:00:5e:00:53:01
  Last flapped   : 2015-05-12 08:36:59 UTC (1w2d 00:00 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
  Egress queues: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                    0                    0                    0

    1 expedited-fo                   0                    0                    0

    2 assured-forw                   0                    0                    0

    3 network-cont                   0                    0                    0

  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  Active alarms  : LINK
  Active defects : LINK
```

```
Interface transmit statistics: Disabled

Logical interface ge-0/0/1.0 (Index 71) (SNMP ifIndex 514) (Generation 136)
  Flags: Device-Down 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
  Security: Zone: public
  Flow Statistics :
  Flow Input statistics :
    Self packets :                   0
    ICMP packets :                   0
    VPN packets :                    0
    Multicast packets :              0
    Bytes permitted by policy :      0
    Connections established :        0
  Flow Output statistics:
    Multicast packets :              0
    Bytes permitted by policy :      0
  Flow error statistics (Packets dropped due to):
    Address spoofing:                0
    Authentication failed:           0
    Incoming NAT errors:             0
    Invalid zone received packet:    0
    Multiple user authentications:   0
    Multiple incoming NAT:           0
    No parent for a gate:            0
    No one interested in self packets: 0
    No minor session:                0
    No more sessions:                0
    No NAT gate:                     0
    No route present:                0
    No SA for incoming SPI:          0
    No tunnel found:                 0
    No session for a gate:           0
    No zone or NULL zone binding     0
    Policy denied:                   0
    Security association not active:   0
    TCP sequence number out of window: 0
    Syn-attack protection:           0
    User authentication errors:      0
  Protocol inet, MTU: 1500, Generation: 150, Route table: 0
    Flags: Sendbcast-pkt-to-re
    Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
      Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255, Generation:
150
```

show interfaces statistics st0.0 detail

```
user@host> show interfaces statistics st0.0 detail
  Logical interface st0.0 (Index 71) (SNMP ifIndex 609) (Generation 136)
    Flags: Up Point-To-Point SNMP-Traps Encapsulation: Secure-Tunnel
    Traffic statistics:
     Input  bytes  :         528152756774
     Output bytes  :         575950643520
     Input  packets:          11481581669
     Output packets:          12520666095
    Local statistics:
     Input  bytes  :                    0
     Output bytes  :                    0
     Input  packets:                    0
     Output packets:                    0
    Transit statistics:
     Input  bytes  :                    0            121859888 bps
     Output bytes  :                    0            128104112 bps
     Input  packets:                    0               331141 pps
     Output packets:                    0               348108 pps
    Security: Zone: untrust
    Allowed host-inbound traffic : any-service bfd bgp dvmrp igmp ldp msdp nhrp
ospf ospf3 pgm pim rip ripng router-discovery rsvp
    sap vrrp
    Flow Statistics :
    Flow Input statistics :
      Self packets :                   0
      ICMP packets :                   0
      VPN packets :                    0
      Multicast packets :              0
      Bytes permitted by policy :      525984295844
      Connections established :        7
    Flow Output statistics:
      Multicast packets :              0
      Bytes permitted by policy :      576003290222
    Flow error statistics (Packets dropped due to):
      Address spoofing:                0
      Authentication failed:           0
      Incoming NAT errors:             0
      Invalid zone received packet:    0
      Multiple user authentications:   0
      Multiple incoming NAT:           0
      No parent for a gate:            0
      No one interested in self packets: 0
      No minor session:                0
      No more sessions:                0
      No NAT gate:                     0
      No route present:                2000280
      No SA for incoming SPI:          0
      No tunnel found:                 0
      No session for a gate:           0
      No zone or NULL zone binding     0
      Policy denied:                   0
      Security association not active:   0
      TCP sequence number out of window: 0
      Syn-attack protection:           0
      User authentication errors:      0
    Protocol inet, MTU: 9192
    Max nh cache: 0, New hold nh limit: 0, Curr nh cnt: 0, Curr new hold cnt: 0,
 NH drop cnt: 0
    Generation: 155, Route table: 0
```

```
    Flags: Sendbcast-pkt-to-re
```

show interfaces extensive (Gigabit Ethernet)

```
user@host> show interfaces ge-0/0/1.0 extensive
Physical interface: ge-0/0/1, Enabled, Physical link is Down
  Interface index: 135, SNMP ifIndex: 510, Generation: 138
  Link-level type: Ethernet, MTU: 1514, Link-mode: Full-duplex, Speed: 1000mbps,

  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online
  Device flags   : Present Running Down
  Interface flags: Hardware-Down SNMP-Traps Internal: 0x0
  Link flags     : None
  CoS queues     : 8 supported, 8 maximum usable queues
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 00:00:5e:00:53:01, Hardware address: 00:00:5e:00:53:01
  Last flapped   : 2015-05-12 08:36:59 UTC (1w1d 22:57 ago)
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                    0                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                    0                    0 pps
   Output packets:                    0                    0 pps
  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: 8 supported, 4 in use
  Queue counters:       Queued packets  Transmitted packets      Dropped packets

    0 best-effort                   0                    0                    0

    1 expedited-fo                  0                    0                    0

    2 assured-forw                  0                    0                    0

    3 network-cont                  0                    0                    0

  Queue number:         Mapped forwarding classes
    0                   best-effort
    1                   expedited-forwarding
    2                   assured-forwarding
    3                   network-control
  Active alarms  : LINK
  Active defects : LINK
  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
  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: 2, CAM source filters: 0
  Autonegotiation information:
    Negotiation status: Incomplete
  Packet Forwarding Engine configuration:
    Destination slot: 0
  CoS information:
    Direction : Output
    CoS transmit queue               Bandwidth               Buffer Priority
Limit
                              %            bps     %           usec
    0 best-effort            95      950000000    95              0      low
none
    3 network-control         5       50000000     5              0      low
none
  Interface transmit statistics: Disabled

  Logical interface ge-0/0/1.0 (Index 71) (SNMP ifIndex 514) (Generation 136)
    Flags: Device-Down 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
    Security: Zone: public
    Flow Statistics :
    Flow Input statistics :
      Self packets :                   0
      ICMP packets :                   0
      VPN packets :                    0
      Multicast packets :              0
      Bytes permitted by policy :      0
      Connections established :        0
    Flow Output statistics:
      Multicast packets :              0
      Bytes permitted by policy :      0
    Flow error statistics (Packets dropped due to):
      Address spoofing:                0
```

```
    Authentication failed:             0
    Incoming NAT errors:               0
    Invalid zone received packet:      0
    Multiple user authentications:     0
    Multiple incoming NAT:             0
    No parent for a gate:              0
    No one interested in self packets: 0
    No minor session:                  0
    No more sessions:                  0
    No NAT gate:                       0
    No route present:                  0
    No SA for incoming SPI:            0
    No tunnel found:                   0
    No session for a gate:             0
    No zone or NULL zone binding       0
    Policy denied:                     0
    Security association not active:   0
    TCP sequence number out of window: 0
    Syn-attack protection:             0
    User authentication errors:        0
  Protocol inet, MTU: 1500, Generation: 150, Route table: 0
    Flags: Sendbcast-pkt-to-re
    Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
      Destination: 1.1.1/24, Local: 1.1.1.1, Broadcast: 1.1.1.255,
      Generation: 150
```

show interfaces terse

```
user@host> show interfaces terse
    Interface               Admin Link Proto    Local                 Remote
    ge-0/0/0                up    up
    ge-0/0/0.0              up    up   inet     10.209.4.61/18
    gr-0/0/0                up    up
    ip-0/0/0                up    up
    st0                     up    up
    st0.1                   up    ready inet
    ls-0/0/0                up    up
    lt-0/0/0                up    up
    mt-0/0/0                up    up
    pd-0/0/0                up    up
    pe-0/0/0                up    up
    e3-1/0/0                up    up
    t3-2/0/0                up    up
    e1-3/0/0                up    up
    se-4/0/0                up    down
    t1-5/0/0                up    up
    br-6/0/0                up    up
    dc-6/0/0                up    up
    dc-6/0/0.32767          up    up
    bc-6/0/0:1              down  up
    bc-6/0/0:1.0            up    down
    dl0                     up    up
    dl0.0                   up    up   inet
    dsc                     up    up
    gre                     up    up
    ipip                    up    up
    lo0                     up    up
    lo0.16385               up    up   inet     10.0.0.1              --> 0/0
                                                 10.0.0.16             --> 0/0
```

```
    lsi                    up    up
    mtun                   up    up
    pimd                   up    up
    pime                   up    up
    pp0                    up    up
```

show interfaces controller (Channelized E1 IQ with Logical E1)

```
user@host> show interfaces controller ce1-1/2/6

Controller                                                      Admin Link
ce1-1/2/6                                                       up    up
e1-1/2/6                                                        up    up
```

show interfaces controller (Channelized E1 IQ with Logical DS0)

```
user@host> show interfaces controller ce1-1/2/3

Controller                                                      Admin Link
ce1-1/2/3                                                       up    up

ds-1/2/3:1                                                      up    up

ds-1/2/3:2                                                      up    up
```

show interfaces descriptions

```
user@host> show interfaces descriptions
 Interface       Admin Link Description
 so-1/0/0        up    up   M20-3#1
 so-2/0/0        up    up   GSR-12#1
 ge-3/0/0        up    up   SMB-OSPF_Area300
 so-3/3/0        up    up   GSR-13#1
 so-3/3/1        up    up   GSR-13#2
 ge-4/0/0        up    up   T320-7#1
 ge-5/0/0        up    up   T320-7#2
 so-7/1/0        up    up   M160-6#1
 ge-8/0/0        up    up   T320-7#3
 ge-9/0/0        up    up   T320-7#4
 so-10/0/0       up    up   M160-6#2
 so-13/0/0       up    up   M20-3#2
 so-14/0/0       up    up   GSR-12#2
 ge-15/0/0       up    up   SMB-OSPF_Area100
 ge-15/0/1       up    up   GSR-13#3
```

show interfaces destination-class all

```
user@host> show interfaces destination-class all
Logical interface so-4/0/0.0
                                          Packets                 Bytes
     Destination class            (packet-per-second)    (bits-per-second)
                       gold                        0                    0
                                               (    0) (                 0)
                     silver                        0                    0
                                               (    0) (                 0)
  Logical interface so-0/1/3.0
                                          Packets                 Bytes
     Destination class            (packet-per-second)    (bits-per-second)
                       gold                        0                    0
                                               (    0) (                 0)
```

```
                                  silver                        0                    0
                                            (                   0) (                 0)
```

show interfaces diagnostics optics

```
user@host> show interfaces diagnostics optics ge-2/0/0
Physical interface: ge-2/0/0
    Laser bias current                         :  7.408 mA
    Laser output power                         :  0.3500 mW / -4.56 dBm
    Module temperature                         :  23 degrees C / 73 degrees F
    Module voltage                             :  3.3450 V
    Receiver signal average optical power      :  0.0002 mW / -36.99 dBm
    Laser bias current high alarm              :  Off
    Laser bias current low alarm               :  Off
    Laser bias current high warning            :  Off
    Laser bias current low warning             :  Off
    Laser output power high alarm              :  Off
    Laser output power low alarm               :  Off
    Laser output power high warning            :  Off
    Laser output power low warning             :  Off
    Module temperature high alarm              :  Off
    Module temperature low alarm               :  Off
    Module temperature high warning            :  Off
    Module temperature low warning             :  Off
    Module voltage high alarm                  :  Off
    Module voltage low alarm                   :  Off
    Module voltage high warning                :  Off
    Module voltage low warning                 :  Off
    Laser rx power high alarm                  :  Off
    Laser rx power low alarm                   :  On
    Laser rx power high warning                :  Off
    Laser rx power low warning                 :  On
    Laser bias current high alarm threshold    :  17.000 mA
    Laser bias current low alarm threshold     :  1.000 mA
    Laser bias current high warning threshold  :  14.000 mA
    Laser bias current low warning threshold   :  2.000 mA
    Laser output power high alarm threshold    :  0.6310 mW / -2.00 dBm
    Laser output power low alarm threshold     :  0.0670 mW / -11.74 dBm
    Laser output power high warning threshold  :  0.6310 mW / -2.00 dBm
    Laser output power low warning threshold   :  0.0790 mW / -11.02 dBm
    Module temperature high alarm threshold    :  95 degrees C / 203 degrees F
    Module temperature low alarm threshold     :  -25 degrees C / -13 degrees F
    Module temperature high warning threshold  :  90 degrees C / 194 degrees F
    Module temperature low warning threshold   :  -20 degrees C / -4 degrees F
    Module voltage high alarm threshold        :  3.900 V
    Module voltage low alarm threshold         :  2.700 V
    Module voltage high warning threshold      :  3.700 V
    Module voltage low warning threshold       :  2.900 V
    Laser rx power high alarm threshold        :  1.2590 mW / 1.00 dBm
    Laser rx power low alarm threshold         :  0.0100 mW / -20.00 dBm
    Laser rx power high warning threshold      :  0.7940 mW / -1.00 dBm
    Laser rx power low warning threshold       :  0.0158 mW / -18.01 dBm
```

show interfaces far-end-interval coc12-5/2/0

```
user@host> show interfaces far-end-interval coc12-5/2/0
Physical interface: coc12-5/2/0, SNMP ifIndex: 121
  05:30-current:
    ES-L: 1, SES-L: 1, UAS-L: 0
```

```
  05:15-05:30:
    ES-L: 0, SES-L: 0, UAS-L: 0
  05:00-05:15:
    ES-L: 0, SES-L: 0, UAS-L: 0
  04:45-05:00:
    ES-L: 0, SES-L: 0, UAS-L: 0
  04:30-04:45:
    ES-L: 0, SES-L: 0, UAS-L: 0
  04:15-04:30:
    ES-L: 0, SES-L: 0, UAS-L: 0
  04:00-04:15:
...
```

show interfaces far-end-interval coc1-5/2/1:1

```
user@host> run show interfaces far-end-interval coc1-5/2/1:1
 Physical interface: coc1-5/2/1:1, SNMP ifIndex: 342
  05:30-current:
    ES-L: 1, SES-L: 1, UAS-L: 0, ES-P: 0, SES-P: 0, UAS-P: 0
  05:15-05:30:
    ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0, SES-P: 0, UAS-P: 0
  05:00-05:15:
    ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0, SES-P: 0, UAS-P: 0
  04:45-05:00:
    ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0, SES-P: 0, UAS-P: 0
  04:30-04:45:
    ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0, SES-P: 0, UAS-P: 0
  04:15-04:30:
    ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0, SES-P: 0, UAS-P: 0
  04:00-04:15:
```

show interfaces filters

```
user@host> show interfaces filters
Interface       Admin Link Proto Input Filter         Output Filter
ge-0/0/0        up    up
ge-0/0/0.0      up    up   inet
                           iso
ge-5/0/0        up    up
ge-5/0/0.0      up    up   any                        f-any
                           inet                       f-inet
                           multiservice
gr-0/3/0        up    up
ip-0/3/0        up    up
mt-0/3/0        up    up
pd-0/3/0        up    up
pe-0/3/0        up    up
vt-0/3/0        up    up
at-1/0/0        up    up
at-1/0/0.0      up    up   inet
                           iso
at-1/1/0        up    down
at-1/1/0.0      up    down inet
                           iso
....
```

show interfaces flow-statistics (Gigabit Ethernet)

```
user@host> show interfaces flow-statistics ge-0/0/1.0
```

```
  Logical interface ge-0/0/1.0 (Index 70) (SNMP ifIndex 49)
    Flags: SNMP-Traps Encapsulation: ENET2
    Input packets : 5161
    Output packets: 83
    Security: Zone: zone2
    Allowed host-inbound traffic : bootp bfd bgp  dns dvmrp  ldp msdp nhrp ospf
pgm
    pim rip router-discovery rsvp sap vrrp dhcp finger ftp tftp ident-reset http
 https ike
   netconf ping rlogin rpm rsh snmp snmp-trap ssh telnet traceroute xnm-clear-text
 xnm-ssl
    lsping
    Flow Statistics :
    Flow Input statistics :
      Self packets :                    0
      ICMP packets :                    0
      VPN packets :                     2564
      Bytes permitted by policy :       3478
      Connections established :         1
    Flow Output statistics:
      Multicast packets :               0
      Bytes permitted by policy :       16994
    Flow error statistics (Packets dropped due to):
      Address spoofing:                 0
      Authentication failed:            0
      Incoming NAT errors:              0
      Invalid zone received packet:     0
      Multiple user authentications:    0
      Multiple incoming NAT:            0
      No parent for a gate:             0
      No one interested in self packets: 0
      No minor session:                 0
      No more sessions:                 0
      No NAT gate:                      0
      No route present:                 0
      No SA for incoming SPI:           0
      No tunnel found:                  0
      No session for a gate:            0
      No zone or NULL zone binding      0
      Policy denied:                    0
      Security association not active:  0
      TCP sequence number out of window: 0
      Syn-attack protection:            0
      User authentication errors:       0
    Protocol inet, MTU: 1500
      Flags: None
      Addresses, Flags: Is-Preferred Is-Primary
        Destination:  203.0.113.1/24, Local: 203.0.113.2, Broadcast: 2.2.2.255
```

show interfaces interval (Channelized OC12)

```
user@host> show interfaces interval t3-0/3/0:0
Physical interface: t3-0/3/0:0, SNMP ifIndex: 23
  17:43-current:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  17:28-17:43:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  17:13-17:28:
```

```
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  16:58-17:13:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  16:43-16:58:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
...
Interval Total:
    LCV: 230, PCV: 1145859, CCV: 455470, LES: 0, PES: 230, PSES: 230,
    CES: 230, CSES: 230, SEFS: 230, UAS: 238
```

show interfaces interval (E3)

```
user@host> show interfaces interval e3-0/3/0
Physical interface: e3-0/3/0, SNMP ifIndex: 23
  17:43-current:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  17:28-17:43:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  17:13-17:28:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  16:58-17:13:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
    SEFS: 0, UAS: 0
  16:43-16:58:
    LCV: 0, PCV: 0, CCV: 0, LES: 0, PES: 0, PSES: 0, CES: 0, CSES: 0,
....
Interval Total:
    LCV: 230, PCV: 1145859, CCV: 455470, LES: 0, PES: 230, PSES: 230,
    CES: 230, CSES: 230, SEFS: 230, UAS: 238
```

show interfaces interval (SONET/SDH) (SRX devices)

```
user@host> show interfaces interval so-0/1/0
Physical interface: so-0/1/0, SNMP ifIndex: 19
  20:02-current:
    ES-S: 0, SES-S: 0, SEFS-S: 0, ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0,
    SES-P: 0, UAS-P: 0
  19:47-20:02:
    ES-S: 267, SES-S: 267, SEFS-S: 267, ES-L: 267, SES-L: 267, UAS-L: 267,
    ES-P: 267, SES-P: 267, UAS-P: 267
  19:32-19:47:
    ES-S: 56, SES-S: 56, SEFS-S: 56, ES-L: 56, SES-L: 56, UAS-L: 46, ES-P: 56,
    SES-P: 56, UAS-P: 46
  19:17-19:32:
    ES-S: 0, SES-S: 0, SEFS-S: 0, ES-L: 0, SES-L: 0, UAS-L: 0, ES-P: 0,
    SES-P: 0, UAS-P: 0
  19:02-19:17:
.....
```

show interfaces load-balancing (SRX devices)

```
user@host> show interfaces load-balancing
Interface  State           Last change  Member count
ams0       Up              1d 00:50     2
ams1       Up              00:00:59     2
```

show interfaces load-balancing detail (SRX devices)

```
user@host>show interfaces load-balancing detail
Load-balancing interfaces detail
Interface        : ams0
  State          : Up
  Last change    : 1d 00:51
  Member count   : 2
  Members        :
      Interface    Weight   State
      mams-2/0/0    10        Active
      mams-2/1/0    10        Active
```

show interfaces mac-database (All MAC Addresses on a Port SRX devices)

```
user@host>  show interfaces mac-database xe-0/3/3
Physical interface: xe-0/3/3, Enabled, Physical link is Up
  Interface index: 372, SNMP ifIndex: 788
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, Loopback:
None, Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface xe-0/3/3.0 (Index 364) (SNMP ifIndex 829)
    Flags: SNMP-Traps 0x4004000 Encapsulation: ENET2
  MAC address        Input frames    Input bytes  Output frames   Output bytes
  00:00:00:00:00:00             1             56              0              0
  00:00:c0:01:01:02       7023810      323095260              0              0
  00:00:c0:01:01:03       7023810      323095260              0              0
  00:00:c0:01:01:04       7023810      323095260              0              0
  00:00:c0:01:01:05       7023810      323095260              0              0
  00:00:c0:01:01:06       7023810      323095260              0              0
  00:00:c0:01:01:07       7023810      323095260              0              0
  00:00:c0:01:01:08       7023809      323095214              0              0
  00:00:c0:01:01:09       7023809      323095214              0              0
  00:00:c0:01:01:0a       7023809      323095214              0              0
  00:00:c0:01:01:0b       7023809      323095214              0              0
  00:00:c8:01:01:02      30424784     1399540064       37448598     1722635508
  00:00:c8:01:01:03      30424784     1399540064       37448598     1722635508
  00:00:c8:01:01:04      30424716     1399536936       37448523     1722632058
  00:00:c8:01:01:05      30424789     1399540294       37448598     1722635508
  00:00:c8:01:01:06      30424788     1399540248       37448597     1722635462
  00:00:c8:01:01:07      30424783     1399540018       37448597     1722635462
  00:00:c8:01:01:08      30424783     1399540018       37448596     1722635416
  00:00:c8:01:01:09       8836796      406492616        8836795      406492570
  00:00:c8:01:01:0a      30424712     1399536752       37448521     1722631966
  00:00:c8:01:01:0b      30424715     1399536890       37448523     1722632058
  Number of MAC addresses : 21
```

show interfaces mac-database (All MAC Addresses on a Service SRX devices)

```
user@host> show interfaces mac-database xe-0/3/3
Logical interface xe-0/3/3.0 (Index 364) (SNMP ifIndex 829)
    Flags: SNMP-Traps 0x4004000 Encapsulation: ENET2
  MAC address        Input frames    Input bytes  Output frames   Output bytes
  00:00:00:00:00:00             1             56              0              0
```

```
  00:00:c0:01:01:02       7023810       323095260              0              0
  00:00:c0:01:01:03       7023810       323095260              0              0
  00:00:c0:01:01:04       7023810       323095260              0              0
  00:00:c0:01:01:05       7023810       323095260              0              0
  00:00:c0:01:01:06       7023810       323095260              0              0
  00:00:c0:01:01:07       7023810       323095260              0              0
  00:00:c0:01:01:08       7023809       323095214              0              0
  00:00:c0:01:01:09       7023809       323095214              0              0
  00:00:c0:01:01:0a       7023809       323095214              0              0
  00:00:c0:01:01:0b       7023809       323095214              0              0
  00:00:c8:01:01:02      31016568      1426762128       38040381     1749857526
  00:00:c8:01:01:03      31016568      1426762128       38040382     1749857572
  00:00:c8:01:01:04      31016499      1426758954       38040306     1749854076
  00:00:c8:01:01:05      31016573      1426762358       38040381     1749857526
  00:00:c8:01:01:06      31016573      1426762358       38040381     1749857526
  00:00:c8:01:01:07      31016567      1426762082       38040380     1749857480
  00:00:c8:01:01:08      31016567      1426762082       38040379     1749857434
  00:00:c8:01:01:09       9428580       433714680        9428580      433714680
  00:00:c8:01:01:0a      31016496      1426758816       38040304     1749853984
  00:00:c8:01:01:0b      31016498      1426758908       38040307     1749854122
```

show interfaces mac-database mac-address

```
user@host> show interfaces mac-database xe-0/3/3 mac-address (SRX devices)
00:00:c8:01:01:09
Physical interface: xe-0/3/3, Enabled, Physical link is Up
  Interface index: 372, SNMP ifIndex: 788
  Link-level type: Ethernet, MTU: 1514, LAN-PHY mode, Speed: 10Gbps, Loopback:
None, Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags     : None

  Logical interface xe-0/3/3.0 (Index 364) (SNMP ifIndex 829)
    Flags: SNMP-Traps 0x4004000 Encapsulation: ENET2
  MAC address: 00:00:c8:01:01:09, Type: Configured,
    Input bytes     : 202324652
    Output bytes    : 202324560
    Input frames    : 4398362
    Output frames   : 4398360
  Policer statistics:
  Policer type      Discarded frames    Discarded bytes
  Output aggregate           3992386          183649756
```

show interfaces mc-ae (SRX devices)

```
user@host> show interfaces mc-ae ae0 unit 512
  Member Links   : ae0
  Local Status   : active
  Peer Status    : active
  Logical Interface        : ae0.512
  Core Facing Interface : Label Ethernet Interface
  ICL-PL         : Label Ethernet Interface
```

show interfaces media (SONET/SDH)

The following example displays the output fields unique to the **show interfaces media** command for a SONET interface (with no level of output specified):

```
user@host> show interfaces media so-4/1/2
Physical interface: so-4/1/2, Enabled, Physical link is Up
  Interface index: 168, SNMP ifIndex: 495
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC48,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running
  Interface flags: Point-To-Point SNMP-Traps 16384
  Link flags     : Keepalives
  Keepalive settings: Interval 10 seconds, Up-count 1, Down-count 3
  Keepalive: Input: 1783 (00:00:00 ago), Output: 1786 (00:00:08 ago)
  LCP state: Opened
  NCP state: inet: Not-configured, inet6: Not-configured, iso: Not-configured,
  mpls: Not-configured
  CHAP state: Not-configured
  CoS queues     : 8 supported
  Last flapped   : 2005-06-15 12:14:59 PDT (04:31:29 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SONET alarms   : None
  SONET defects  : None
  SONET errors:
     BIP-B1: 121, BIP-B2: 916, REI-L: 0, BIP-B3: 137, REI-P: 16747, BIP-BIP2: 0
  Received path trace: routerb so-1/1/2
  Transmitted path trace: routera so-4/1/2
```

show interfaces policers (SRX devices)

```
user@host> show interfaces policers
Interface       Admin Link Proto Input Policer         Output Policer
ge-0/0/0        up    up
ge-0/0/0.0      up    up   inet
                           iso
gr-0/3/0        up    up
ip-0/3/0        up    up
mt-0/3/0        up    up
pd-0/3/0        up    up
pe-0/3/0        up    up
...
so-2/0/0        up    up
so-2/0/0.0      up    up   inet  so-2/0/0.0-in-policer so-2/0/0.0-out-policer
                           iso
so-2/1/0        up    down
...
```

show interfaces policers interface-name (SRX devices)

```
user@host> show interfaces policers so-2/1/0
Interface       Admin Link Proto Input Policer         Output Policer
so-2/1/0        up    down
so-2/1/0.0      up    down inet  so-2/1/0.0-in-policer so-2/1/0.0-out-policer
                           iso
                           inet6
```

show interfaces queue (SRX devices)

The following truncated example shows the CoS queue sizes for queues 0, 1, and 3. Queue 1 has a queue buffer size (guaranteed allocated memory) of 9192 bytes.

```
user@host> show interfaces queue
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 134, SNMP ifIndex: 509
Forwarding classes: 8 supported, 8 in use
Egress queues: 8 supported, 8 in use
Queue: 0, Forwarding classes: class0
  Queued:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
  Transmitted:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
    Tail-dropped packets :                     0                     0 pps
    RL-dropped packets   :                     0                     0 pps
    RL-dropped bytes     :                     0                     0 bps
    RED-dropped packets  :                     0                     0 pps
     Low                 :                     0                     0 pps
     Medium-low          :                     0                     0 pps
     Medium-high         :                     0                     0 pps
     High                :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
     Low                 :                     0                     0 bps
     Medium-low          :                     0                     0 bps
     Medium-high         :                     0                     0 bps
     High                :                     0                     0 bps
  Queue Buffer Usage:
    Reserved buffer      :             118750000 bytes
  Queue-depth bytes      :
    Current              :                     0
..
..
Queue: 1, Forwarding classes: class1
..
..
 Queue Buffer Usage:
    Reserved buffer      :                  9192 bytes
  Queue-depth bytes      :
    Current              :                     0
..
..
Queue: 3, Forwarding classes: class3
  Queued:
..
..
Queue Buffer Usage:
    Reserved buffer      :               6250000 bytes
  Queue-depth bytes      :
    Current              :                     0
..
..
```

show interfaces redundancy (SRX devices)

```
user@host> show interfaces redundancy
Interface  State         Last change  Primary    Secondary  Current status
rsp0       Not present                sp-1/0/0   sp-0/2/0   both down
rsp1       On secondary  1d 23:56     sp-1/2/0   sp-0/3/0   primary down
rsp2       On primary    10:10:27     sp-1/3/0   sp-0/2/0   secondary down
rlsq0      On primary    00:06:24     lsq-0/3/0  lsq-1/0/0  both up
```

show interfaces redundancy (Aggregated Ethernet SRX devices)

```
user@host> show interfaces redundancy
Interface  State           Last change  Primary      Secondary    Current status
rlsq0      On secondary     00:56:12     lsq-4/0/0    lsq-3/0/0    both up

ae0
ae1
ae2
ae3
ae4
```

show interfaces redundancy detail (SRX devices)

```
user@host> show interfaces redundancy detail
Interface        : rlsq0
  State          : On primary
  Last change    : 00:45:47
  Primary        : lsq-0/2/0
  Secondary      : lsq-1/2/0
  Current status : both up
  Mode           : hot-standby

Interface        : rlsq0:0
  State          : On primary
  Last change    : 00:45:46
  Primary        : lsq-0/2/0:0
  Secondary      : lsq-1/2/0:0
  Current status : both up
  Mode           : warm-standby
```

show interfaces routing brief (SRX devices)

```
user@host> show interfaces routing brief
Interface        State Addresses
so-5/0/3.0       Down  ISO   enabled
so-5/0/2.0       Up    MPLS  enabled
                       ISO   enabled
                       INET  192.168.2.120
                       INET  enabled
so-5/0/1.0       Up    MPLS  enabled
                       ISO   enabled
                       INET  192.168.2.130
                       INET  enabled
at-1/0/0.3       Up    CCC   enabled
at-1/0/0.2       Up    CCC   enabled
at-1/0/0.0       Up    ISO   enabled
                       INET  192.168.90.10
                       INET  enabled
lo0.0            Up    ISO   47.0005.80ff.f800.0000.0108.0001.1921.6800.5061.00
                       ISO   enabled
                       INET  127.0.0.1
fxp1.0           Up
fxp0.0           Up    INET  192.168.6.90
```

show interfaces routing detail (SRX devices)

```
user@host> show interfaces routing detail
so-5/0/3.0
  Index: 15, Refcount: 2, State: Up <Broadcast PointToPoint Multicast> Change:<>
```

```
  Metric: 0, Up/down transitions: 0, Full-duplex
  Link layer: HDLC serial line Encapsulation: PPP Bandwidth: 155Mbps
  ISO address (null)
    State: <Broadcast PointToPoint Multicast> Change: <>
    Preference: 0 (120 down), Metric: 0, MTU: 4470 bytes
so-5/0/2.0
  Index: 14, Refcount: 7, State: <Up Broadcast PointToPoint Multicast> Change:<>

  Metric: 0, Up/down transitions: 0, Full-duplex
  Link layer: HDLC serial line Encapsulation: PPP Bandwidth: 155Mbps
  MPLS address (null)
    State: <Up Broadcast PointToPoint Multicast> Change: <>
    Preference: 0 (120 down), Metric: 0, MTU: 4458 bytes
  ISO address (null)
    State: <Up Broadcast PointToPoint Multicast> Change: <>
    Preference: 0 (120 down), Metric: 0, MTU: 4470 bytes
  INET address 192.168.2.120
    State: <Up Broadcast PointToPoint Multicast Localup> Change: <>
    Preference: 0 (120 down), Metric: 0, MTU: 4470 bytes
    Local address: 192.168.2.120
    Destination: 192.168.2.110/32
  INET address (null)
    State: <Up Broadcast PointToPoint Multicast> Change: <>
    Preference: 0 (120 down), Metric: 0, MTU: 4470 bytes
...
```

show interfaces routing-instance all (SRX devices)

```
user@host> show interfaces terse routing-instance all
Interface   Admin   Link   Proto    Local             Remote Instance
at-0/0/1    up      up     inet     10.0.0.1/24
ge-0/0/0.0  up      up     inet      192.168.4.28/24          sample-a
at-0/1/0.0  up      up     inet6     fe80::a:0:0:4/64         sample-b
so-0/0/0.0  up      up     inet     10.0.0.1/32
```

show interfaces snmp-index (SRX devices)

```
user@host> show interfaces snmp-index 33
Physical interface: so-2/1/1, Enabled, Physical link is Down
  Interface index: 149, SNMP ifIndex: 33
  Link-level type: PPP, MTU: 4474, Clocking: Internal, SONET mode, Speed: OC48,
  Loopback: None, FCS: 16, Payload scrambler: Enabled
  Device flags   : Present Running Down
  Interface flags: Hardware-Down Point-To-Point SNMP-Traps 16384
  Link flags     : Keepalives
  CoS queues     : 8 supported
  Last flapped   : 2005-06-15 11:45:57 PDT (05:38:43 ago)
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  SONET alarms   : LOL, PLL, LOS
  SONET defects  : LOL, PLL, LOF, LOS, SEF, AIS-L, AIS-P
```

show interfaces source-class all (SRX devices)

```
user@host> show interfaces source-class all
Logical interface so-0/1/0.0
                                                Packets                Bytes
      Source class                    (packet-per-second)    (bits-per-second)
                              gold              1928095             161959980
```

```
                                        (                889) (             597762)
                                bronze                     0                    0
                                        (                  0) (                  0)
                                silver                     0                    0
                                        (                  0) (                  0)
  Logical interface so-0/1/3.0
                                                     Packets                Bytes
      Source class                    (packet-per-second)    (bits-per-second)
                                  gold                     0                    0
                                        (                  0) (                  0)
                                bronze                     0                    0
                                        (                  0) (                  0)
                                silver                116113              9753492
                                        (                939) (             631616)
```

show interfaces statistics (Fast Ethernet SRX devices)

```
user@host> show interfaces fe-1/3/1 statistics
Physical interface: fe-1/3/1, Enabled, Physical link is Up
  Interface index: 144, SNMP ifIndex: 1042
  Description: ford fe-1/3/1
  Link-level type: Ethernet, MTU: 1514, Speed: 100mbps, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Enabled
  Device flags   : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  CoS queues     : 4 supported, 4 maximum usable queues
  Current address: 00:90:69:93:04:dc, Hardware address: 00:90:69:93:04:dc
  Last flapped   : 2006-04-18 03:08:59 PDT (00:01:24 ago)
  Statistics last cleared: Never
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
  Input errors: 0, Output errors: 0
  Active alarms  : None
  Active defects : None
  Logical interface fe-1/3/1.0 (Index 69) (SNMP ifIndex 50)
    Flags: SNMP-Traps Encapsulation: ENET2
    Protocol inet, MTU: 1500
      Flags: Is-Primary, DCU, SCU-in
                                                 Packets                Bytes
      Destination class                 (packet-per-second)    (bits-per-second)
                              silver1                    0                    0
                                        (                  0) (                  0)
                              silver2                    0                    0
                                        (                  0) (                  0)
                              silver3                    0                    0
                                        (                  0) (                  0)
      Addresses, Flags: Is-Default Is-Preferred Is-Primary
        Destination: 10.27.245/24, Local: 10.27.245.2,
        Broadcast: 10.27.245.255
    Protocol iso, MTU: 1497
      Flags: Is-Primary
```

show interfaces switch-port (SRX devices)

```
user@host# show interfaces ge-slot/0/0 switch-port port-number
Port 0, Physical link is Up
        Speed: 100mbps, Auto-negotiation: Enabled
   Statistics:                          Receive         Transmit
          Total bytes                    28437086         21792250
          Total packets                    409145            88008
```

```
          Unicast packets                        9987            83817
          Multicast packets                    145002                0
          Broadcast packets                    254156             4191
          Multiple collisions                      23               10
          FIFO/CRC/Align errors                     0                0
          MAC pause frames                          0                0
          Oversized frames                          0
          Runt frames                               0
          Jabber frames                             0
          Fragment frames                           0
          Discarded frames                          0
        Autonegotiation information:
          Negotiation status: Complete
          Link partner:
              Link mode: Full-duplex, Flow control: None, Remote fault: OK, Link
 partner Speed: 100 Mbps
          Local resolution:
              Flow control: None, Remote fault: Link OK
```

show interfaces transport pm (SRX devices)

```
user@host> show interfaces transport pm all current et-0/1/0
Physical interface: et-0/1/0, SNMP ifIndex 515
  14:45-current          Elapse time:900 Seconds
  Near End       Suspect Flag:False             Reason:None
    PM                  COUNT          THRESHOLD       TCA-ENABLED     TCA-RAISED

  OTU-BBE                 0                800              No              No
  OTU-ES                  0                135              No              No
  OTU-SES                 0                90               No              No
  OTU-UAS                 427              90               No              No
  Far End        Suspect Flag:True              Reason:Unknown
    PM                  COUNT          THRESHOLD       TCA-ENABLED     TCA-RAISED

  OTU-BBE                 0                800              No              No
  OTU-ES                  0                135              No              No
  OTU-SES                 0                90               No              No
  OTU-UAS                 0                90               No              No
  Near End       Suspect Flag:False             Reason:None
    PM                  COUNT          THRESHOLD       TCA-ENABLED     TCA-RAISED

  ODU-BBE                 0                800              No              No
  ODU-ES                  0                135              No              No
  ODU-SES                 0                90               No              No
  ODU-UAS                 427              90               No              No
  Far End        Suspect Flag:True              Reason:Unknown
    PM                  COUNT          THRESHOLD       TCA-ENABLED     TCA-RAISED

  ODU-BBE                 0                800              No              No
  ODU-ES                  0                135              No              No
  ODU-SES                 0                90               No              No
  ODU-UAS                 0                90               No              No
  FEC            Suspect Flag:False             Reason:None
    PM                  COUNT          THRESHOLD       TCA-ENABLED     TCA-RAISED

  FEC-CorrectedErr        2008544300       0                NA              NA
  FEC-UncorrectedWords    0                0                NA              NA
  BER            Suspect Flag:False             Reason:None
    PM                 MIN       MAX       AVG       THRESHOLD       TCA-ENABLED
     TCA-RAISED
  BER                 3.6e-5    5.8e-5    3.6e-5     10.0e-3         No
```

```
      Yes
Physical interface: et-0/1/0, SNMP ifIndex 515
  14:45-current
  Suspect Flag:True             Reason:Object Disabled
    PM                           CURRENT   MIN       MAX       AVG       THRESHOLD
      TCA-ENABLED       TCA-RAISED
                                                                                (MIN)
(MAX)    (MIN) (MAX)       (MIN) (MAX)
  Lane chromatic dispersion          0         0         0         0         0
 0         NA    NA          NA     NA
  Lane differential group delay      0         0         0         0         0
 0         NA    NA          NA     NA
    q Value                             120       120       120       120       0
   0         NA    NA          NA     NA
  SNR                                  28        28        29        28        0
 0         NA    NA          NA     NA
  Tx output power(0.01dBm)           -5000     -5000     -5000     -5000     -300
 -100      No    No          No     No
  Rx input power(0.01dBm)            -3642     -3665     -3626     -3637     -1800
 -500      No    No          No     No
  Module temperature(Celsius)        46        46        46        46        -5
 75        No    No          No     No
  Tx laser bias current(0.1mA)       0         0         0         0         0
 0         NA    NA          NA     NA
  Rx laser bias current(0.1mA)       1270      1270      1270      1270      0
 0         NA    NA          NA     NA
  Carrier frequency offset(MHz)      -186      -186      -186      -186      -5000
 5000      No    No          No     No
```

show security zones (SRX devices)

```
user@host> show security zones
Functional zone: management
   Description: This is the management zone.
   Policy configurable: No
   Interfaces bound: 1
   Interfaces:
     ge-0/0/0.0
Security zone: Host
   Description: This is the host zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     fxp0.0
Security zone: abc
   Description: This is the abc zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     ge-0/0/1.0
Security zone: def
   Description: This is the def zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     ge-0/0/2.0
```

show interfaces irb

Syntax

show interfaces irb
<brief | detail | extensive | terse>
<descriptions>
<media>
<routing-instance *instance-name*>
<snmp-index *snmp-index*>
<statistics>

Release Information

Command introduced in Junos OS Release 12.3R2.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Command introduced in Junos OS Release 13.2 for the QFX Series

Description

Display integrated routing and bridging interfaces information.

Options

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 *instance-name*—(Optional) Display information for the interface with the specified SNMP index.

snmp-index *snmp-index*—(Optional) Display information for the interface with the specified SNMP index.

statistics—(Optional) Display static interface statistics.

Additional Information

Integrated routing and bridging (IRB) provides simultaneous support for Layer 2 bridging and Layer 3 IP routing on the same interface. IRB enables you to route local packets to another routed interface or to another VLAN that has a Layer 3 protocol configured.

Required Privilege Level

view

List of Sample Output

show interfaces irb extensive on page 1348
show interfaces irb snmp-index on page 1349

Output Fields

Table 149 on page 1344 lists the output fields for the **show interfaces irb** command. Output fields are listed in the approximate order in which they appear.

Table 149: show interfaces irb Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	All levels

Table 149: show interfaces irb Output Fields (continued)

Field Name	Field Description	Level of Output
Enabled	State of the physical interface. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.	All levels
Proto	Protocol configured on the interface.	**terse**
Interface index	Physical interface index number, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Type	Physical interface type.	**detail extensive** none
Link-level type	Encapsulation being used on the physical interface.	**detail extensive brief** none
MTU	MTU size on the physical interface.	**detail extensive brief** none
Clocking	Reference clock source: **Internal** or **External**. Always unspecified on IRB interfaces.	**detail extensive brief**
Speed	Speed at which the interface is running. Always unspecified on IRB interfaces.	**detail extensive brief**
Device flags	Information about the physical device. Possible values are described in the "Device Flags" section under *Common Output Fields Description*.	**detail extensive brief** none
Interface flags	Information about the interface. Possible values are described in the "Interface Flags" section under *Common Output Fields Description*.	**detail extensive brief** none
Link type	Physical interface link type: **full duplex** or **half duplex**.	**detail extensive** none
Link flags	Information about the link. Possible values are described in the "Links Flags" section under *Common Output Fields Description*.	**detail extensive** none
Physical Info	Physical interface information.	All levels
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	MAC address of the hardware.	**detail extensive** none
Alternate link address	Backup address of the link.	**detail extensive**
Last flapped	Date, time, and how long ago the interface went from down to up. The format is **Last flapped: year-month-day hours:minutes:seconds timezone (hours:minutes:seconds ago)**. For example, **Last flapped: 2002-04-26 10:52:40 PDT (04:33:20 ago)**.	**detail extensive** none

Table 149: show interfaces irb Output Fields (continued)

Field Name	Field Description	Level of Output
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**
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.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the physical interface if IPv6 statistics tracking is enabled. • **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.	**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. • **Giants**—Number of frames received that are larger than the giant 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 the Junos OS does not handle. • **Resource errors**—Sum of transmit drops.	**detail extensive**
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 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 DPC 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. • **MTU errors**—Number of packets whose size exceeded the MTU of the interface. • **Resource errors**—Sum of transmit drops.	**detail extensive**

Table 149: show interfaces irb Output Fields (continued)

Field Name	Field Description	Level of Output
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 of 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. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	**detail extensive**
Encapsulation	Encapsulation on the logical interface.	**detail extensive**
Bandwidth	Dummy value that is ignored by an IRB interface. IRB interfaces are pseudo interfaces and do not have physical bandwidth associated with them.	**detail extensive**
Routing Instance	Routing instance IRB is configured under.	**detail extensive**
Bridging Domain	Bridging domain IRB is participating in.	**detail extensive**
Traffic statistics	Number and rate of bytes and packets received and transmitted on the logical 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.	**detail extensive**
IPv6 transit statistics	Number of IPv6 transit bytes and packets received and transmitted on the logical interface if IPv6 statistics tracking is enabled. • **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.	**detail extensive**
Local statistics	Statistics for traffic received from and transmitted to the Routing Engine.	**detail extensive**
Transit statistics	Statistics for traffic transiting the router.	**detail extensive**
Protocol	Protocol family configured on the local interface. Possible values are described in the "Protocol Field" section under *Common Output Fields Description*.	**detail extensive**
MTU	Maximum transmission unit size on the logical interface.	**detail extensive**

Table 149: show interfaces irb Output Fields (continued)

Field Name	Field Description	Level of Output
Maximum labels	Maximum number of MPLS labels configured for the MPLS protocol family on the logical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Route table	Routing table in which the logical interface address is located. For example, **0** refers to the routing table inet.0.	**detail extensive**
Addresses, Flags	Information about address flags. Possible values are described in the "Addresses Flags" section under *Common Output Fields Description*.	**detail extensive**
Policer	The policer that is to be evaluated when packets are received or transmitted on the interface.	**detail extensive**
Flags	Information about the logical interface. Possible values are described in the "Logical Interface Flags" section under *Common Output Fields Description*.	**detail extensive**

Sample Output

show interfaces irb extensive

```
user@host> show interfaces irb extensive
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 129, SNMP ifIndex: 23, Generation: 130
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Clocking: Unspecified,
Speed: Unspecified
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Physical info  : Unspecified
  Hold-times     : Up 0 ms, Down 0 ms
  Current address: 02:00:00:00:00:30, Hardware address: 02:00:00:00:00:30
  Alternate link address: Unspecified
  Last flapped   : Never
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :                   0
   Output bytes  :                   0
   Input  packets:                   0
   Output packets:                   0
   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, Giants: 0, Policed discards:
 0, Resource errors: 0
  Output errors:
    Carrier transitions: 0, Errors: 0, Drops: 0, MTU errors: 0, Resource errors:
 0
```

```
  Logical interface irb.0 (Index 68) (SNMP ifIndex 70) (Generation 143)
    Flags: Hardware-Down SNMP-Traps 0x4000 Encapsulation: ENET2
    Bandwidth: 1000mbps
    Routing Instance: customer_0 Bridging Domain: bd0
    Traffic statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
     IPv6 transit 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
     IPv6 transit statistics:
      Input  bytes  :                  0
      Output bytes  :                  0
      Input  packets:                  0
      Output packets:                  0
    Protocol inet, MTU: 1500, Generation: 154, Route table: 0
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.51.1/8, Local: 10.51.1.2, Broadcast: 10.51.1.255,
        Generation: 155
    Protocol multiservice, MTU: 1500, Generation: 155, Route table: 0
      Flags: Is-Primary
      Policer: Input: __default_arp_policer
```

show interfaces irb snmp-index

```
user@host> show interfaces irb snmp-index 25
Physical interface: irb, Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 25
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Link type      : Full-Duplex
  Link flags     : None
  Current address: 02:00:00:00:00:30, Hardware address: 02:00:00:00:00:30
  Last flapped   : Never
    Input packets : 0
    Output packets: 0

  Logical interface irb.0 (Index 68) (SNMP ifIndex 70)
    Flags: Hardware-Down SNMP-Traps 0x4000 Encapsulation: ENET2
    Bandwidth: 1000mbps
    Routing Instance: customer_0 Bridging Domain: bd0
    Input packets : 0
    Output packets: 0
    Protocol inet, MTU: 1500
      Addresses, Flags: Dest-route-down Is-Preferred Is-Primary
        Destination: 10.51.1/8, Local: 10.51.1.2, Broadcast: 10.51.1.255
```

```
Protocol multiservice, MTU: 1500
  Flags: Is-Primary
```

show interfaces queue

Syntax

show interfaces queue
<aggregate | remaining-traffic>
<both-ingress-egress>
<egress>
<forwarding-class *forwarding-class*>
<ingress>
<interface-name *interface-name*>
<l2-statistics>

Release Information

Command introduced before Junos OS Release 7.4.
both-ingress-egress, **egress**, and **ingress** options introduced in Junos OS Release 7.6.
Command introduced in Junos OS Release 11.1 for the QFX Series.
l2-statistics option introduced in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display class-of-service (CoS) queue information for physical interfaces.

Options

none—Show detailed CoS queue statistics for all physical interfaces.

aggregate—(Optional) Display the aggregated queuing statistics of all logical interfaces that have traffic-control profiles configured. (Not on the QFX Series.)

both-ingress-egress—(Optional) On Gigabit Ethernet Intelligent Queuing 2 (IQ2) PICs, display both ingress and egress queue statistics. (Not on the QFX Series.)

egress—(Optional) Display egress queue statistics.

forwarding-class *forwarding-class*—(Optional) Forwarding class name for this queue. Shows detailed CoS statistics for the queue associated with the specified forwarding class.

ingress—(Optional) On Gigabit Ethernet IQ2 PICs, display ingress queue statistics. (Not on the QFX Series.)

interface-name *interface-name*—(Optional) Show detailed CoS queue statistics for the specified interface.

l2-statistics—(Optional) Display Layer 2 statistics for MLPPP, FRF.15, and FRF.16 bundles

remaining-traffic—(Optional) Display the remaining-traffic queue statistics of all logical interfaces that have traffic-control profiles configured.

Overhead for Layer 2 Statistics

Transmitted packets and transmitted byte counts are displayed for the Layer 2 level with the addition of encapsulation overheads applied for fragmentation, as shown in Table 150 on page 1352. Others counters, such as packets and bytes queued (input) and drop counters, are displayed at the Layer 3 level. In the case of link fragmentation

and interleaving (LFI) for which fragmentation is not applied, corresponding Layer 2 overheads are added, as shown in Table 150 on page 1352.

Table 150: Layer 2 Overhead and Transmitted Packets or Byte Counts

Protocol	Fragmentation		LFI
	First fragmentation	Second to *n* fragmentations	
	Bytes	Bytes	
MLPPP (Long)	13	12	8
MLPPP (short)	11	10	8
MLFR (FRF15)	12	10	8
MFR (FRF16)	10	8	-
MCMLPPP(Long)	13	12	-
MCMLPPP(Short)	11	10	-

Layer 2 Statistics—Fragmentation Overhead Calculation

```
MLPPP/MC-MLPPP Overhead details:
   ===============================
   Fragment 1:

    Outer PPP header                        : 4 bytes
    Long or short sequence MLPPP header     : 4 bytes or 2 bytes
    Inner PPP header                        : 1 byte
    HDLC flag and FCS bytes                 : 4 bytes

   Fragments 2 .. n :

    Outer PPP header                        : 4 bytes
    Long or short sequence MLPPP header     : 4 bytes or 2 bytes
    HDLC flag and FCS bytes                 : 4 bytes


   MLFR (FRF15) Overhead details:
   ============================
   Fragment 1:
       Framerelay header     :  2 bytes
       Control,NLPID         :  2 bytes
       Fragmentaion header   :  2 bytes
       Inner proto           :  2 bytes
       HDLC flag and  FCS    :  4 bytes

   Fragments 2 ...n :

       Framerelay header     :  2 bytes
       Control,NLPID         :  2 bytes
       Fragmentaion header   :  2 bytes
       HDLC flag and  FCS    :  4 bytes
```

```
MFR (FRF16) Overhead details:
==============================
Fragment 1:
   Fragmentaion header   :  2 bytes
   Framerelay header     :  2 bytes
   Inner proto           :  2 bytes
   HDLC flag and  FCS    :  4 bytes

Fragments 2 ...n :
   Fragmentaion header   :  2 bytes
   Framerelay header     :  2 bytes
   HDLC flag and  FCS    :  4 bytes
```

Overhead with LFI

```
MLPPP(Long & short sequence):
   ============================
       Outer PPP header      :  4 bytes
       HDLC flag and  FCS    :  4 bytes

  MLFR (FRF15):
  =============
       Framerelay header     :  2 bytes
       Control,NLPID         :  2 bytes
       HDLC flag and  FCS    :  4 bytes
```

The following examples show overhead for different cases:

- A 1000-byte packet is sent to a mlppp bundle without any fragmentation. At the Layer 2 level, bytes transmitted is 1013 in 1 packet. This overhead is for MLPPP long sequence encap.
- A 1000-byte packet is sent to a mlppp bundle with a fragment threshold of 250byte. At the Layer 2 level, bytes transmitted is 1061 bytes in 5 packets.
- A 1000-byte LFI packet is sent to an mlppp bundle. At the Layer 2 level, bytes transmitted is 1008 in 1 packet.

remaining-traffic—(Optional) Display the queuing statistics of all logical interfaces that do not have traffic-control profiles configured. (Not on the QFX Series.)

Additional Information For rate-limited interfaces hosted on Modular Interface Cards (MICs), Modular Port Concentrators (MPCs), or Enhanced Queuing DPCs, rate-limit packet-drop operations occur *before* packets are queued for transmission scheduling. For such interfaces, the statistics for queued traffic do not include the packets that have already been dropped due to rate limiting, and consequently the displayed statistics for queued traffic are the same as the displayed statistics for transmitted traffic.

NOTE: For rate-limited interfaces hosted on other types of hardware, rate-limit packet-drop operations occur *after* packets are queued for transmission scheduling. For these other interface types, the statistics for queued traffic include the packets that are later dropped due to rate limiting, and consequently the displayed statistics for queued traffic equals the sum of the statistics for transmitted and rate-limited traffic.

On M Series routers (except for the M320 and M120 routers), this command is valid only for a PIC installed on an enhanced Flexible PIC Concentrator (FPC).

Queue statistics for aggregated interfaces are supported on the M Series and T Series routers only. Statistics for an aggregated interface are the summation of the queue statistics of the child links of that aggregated interface. You can view the statistics for a child interface by using the **show interfaces statistics** command for that child interface.

When you configure tricolor marking on a 10-port 1-Gigabit Ethernet PIC, for queues 6 and 7 only, the output does not display the number of queued bytes and packets, or the number of bytes and packets dropped because of RED. If you do not configure tricolor marking on the interface, these statistics are available for all queues.

For the 4-port Channelized OC12 IQE PIC and 1-port Channelized OC48 IQE PIC, the **Packet Forwarding Engine Chassis Queues** field represents traffic bound for a particular physical interface on the PIC. For all other PICs, the **Packet Forwarding Engine Chassis Queues** field represents the total traffic bound for the PIC.

For Gigabit Ethernet IQ2 PICs, the **show interfaces queue** command output does not display the number of tail-dropped packets. This limitation does not apply to Packet Forwarding Engine chassis queues.

When fragmentation occurs on the egress interface, the first set of packet counters shows the postfragmentation values. The second set of packet counters (under the **Packet Forwarding Engine Chassis Queues** field) shows the prefragmentation values.

The behavior of the **egress** queues for the **Routing Engine-Generated Traffic** is not same as the configured queue for MLPPP and MFR configurations.

For related CoS operational mode commands, see the CLI Explorer.

Required Privilege Level view

List of Sample Output

show interfaces queue (Rate-Limited Interface on a Gigabit Ethernet MIC in an MPC) on page 1360
show interfaces queue (Aggregated Ethernet on a T320 Router) on page 1361
show interfaces queue (Gigabit Ethernet on a T640 Router) on page 1363
show interfaces queue aggregate (Gigabit Ethernet Enhanced DPC) on page 1363
show interfaces queue (Gigabit Ethernet IQ2 PIC) on page 1367
show interfaces queue both-ingress-egress (Gigabit Ethernet IQ2 PIC) on page 1370

show interfaces queue ingress (Gigabit Ethernet IQ2 PIC) on page 1372
show interfaces queue egress (Gigabit Ethernet IQ2 PIC) on page 1373
show interfaces queue remaining-traffic (Gigabit Ethernet Enhanced DPC) on page 1375
show interfaces queue (Channelized OC12 IQE Type 3 PIC in SONET Mode) on page 1377
show interfaces queue (QFX Series) on page 1387
show interfaces queue l2-statistics (lsq interface) on page 1388
show interfaces queue lsq (lsq-ifd) on page 1389
show interfaces queue (Aggregated Ethernet on a MX series Router) on page 1390

Output Fields Table 151 on page 1355 lists the output fields for the **show interfaces queue** command. Output fields are listed in the approximate order in which they appear.

Table 151: show interfaces queue Output Fields

Field Name	Field Description
Physical interface	Name of the physical interface.
Enabled	State of the interface. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.
Interface index	Physical interface's index number, which reflects its initialization sequence.
SNMP ifIndex	SNMP index number for the interface.
Forwarding classes supported	Total number of forwarding classes supported on the specified interface.
Forwarding classes in use	Total number of forwarding classes in use on the specified interface.
Ingress queues supported	On Gigabit Ethernet IQ2 PICs only, total number of ingress queues supported on the specified interface.
Ingress queues in use	On Gigabit Ethernet IQ2 PICs only, total number of ingress queues in use on the specified interface.
Output queues supported	Total number of output queues supported on the specified interface.
Output queues in use	Total number of output queues in use on the specified interface.
Egress queues supported	Total number of egress queues supported on the specified interface.
Egress queues in use	Total number of egress queues in use on the specified interface.

Table 151: show interfaces queue Output Fields (continued)

Field Name	Field Description
Queue counters (Ingress)	CoS queue number and its associated user-configured forwarding class name. Displayed on IQ2 interfaces. • **Queued packets**—Number of queued packets. NOTE: This field is not supported on QFX5100, QFX5110, QFX5200, and QFX5210 switches due to hardware limitations. • **Transmitted packets**—Number of transmitted packets. • **Dropped packets**—Number of packets dropped by the ASIC's RED mechanism.
Burst size	(Logical interfaces on IQ PICs only) Maximum number of bytes up to which the logical interface can burst. The burst size is based on the shaping rate applied to the interface.
The following output fields are applicable to both interface component and Packet Forwarding component in the **show interfaces queue** command:	
Queue	Queue number.
Forwarding classes	Forwarding class name.
Queued Packets	Number of packets queued to this queue. NOTE: For Gigabit Ethernet IQ2 interfaces, the Queued Packets count is calculated by the Junos OS interpreting one frame buffer as one packet. If the queued packets are very large or very small, the calculation might not be completely accurate for transit traffic. The count is completely accurate for traffic terminated on the router. For rate-limited interfaces hosted on MICs or MPCs only, this statistic does not include traffic dropped due to rate limiting. For more information, see "Additional Information" on page 1353. NOTE: This field is not supported on QFX5100, QFX5110, QFX5200, and QFX5210 switches due to hardware limitations.
Queued Bytes	Number of bytes queued to this queue. The byte counts vary by interface hardware. For more information, see Table 152 on page 1359. For rate-limited interfaces hosted on MICs or MPCs only, this statistic does not include traffic dropped due to rate limiting. For more information, see "Additional Information" on page 1353. NOTE: This field is not supported on QFX5100, QFX5110, QFX5200, and QFX5210 switches due to hardware limitations.
Transmitted Packets	Number of packets transmitted by this queue. When fragmentation occurs on the egress interface, the first set of packet counters shows the postfragmentation values. The second set of packet counters (displayed under the **Packet Forwarding Engine Chassis Queues** field) shows the prefragmentation values. NOTE: For Layer 2 statistics, see "Overhead for Layer 2 Statistics" on page 1351

Table 151: show interfaces queue Output Fields (continued)

Field Name	Field Description
Transmitted Bytes	Number of bytes transmitted by this queue. The byte counts vary by interface hardware. For more information, see Table 152 on page 1359. NOTE: On MX Series routers, this number can be inaccurate when you issue the command for a physical interface repeatedly and in quick succession, because the statistics for the child nodes are collected infrequently. Wait ten seconds between successive iterations to avoid this situation. NOTE: For Layer 2 statistics, see "Overhead for Layer 2 Statistics" on page 1351
Tail-dropped packets	Number of packets dropped because of tail drop. NOTE: Starting with Junos OS 18.3R1, the **Tail-dropped packets** counter is supported on PTX Series Packet Transport Routers.
RL-dropped packets	Number of packets dropped due to rate limiting. For rate-limited interfaces hosted on MICs, MPCs, and Enhanced Queuing DPCs only, this statistic is not included in the queued traffic statistics. For more information, see "Additional Information" on page 1353. NOTE: The **RL-dropped packets** counter is not supported on the PTX Series Packet Transport Routers, and is omitted from the output.
RL-dropped bytes	Number of bytes dropped due to rate limiting. For rate-limited interfaces hosted on MICs, MPCs, and Enhanced Queuing DPCs only, this statistic is not included in the queued traffic statistics. For more information, see "Additional Information" on page 1353.

RED-dropped packets	Number of packets dropped because of random early detection (RED).
	
	• (M Series and T Series routers only) On M320 and M120 routers and the T Series routers, the total number of dropped packets is displayed. On all other M Series routers, the output classifies dropped packets into the following categories:
	  • **Low, non-TCP**—Number of low-loss priority non-TCP packets dropped because of RED.
	  • **Low, TCP**—Number of low-loss priority TCP packets dropped because of RED.
	  • **High, non-TCP**—Number of high-loss priority non-TCP packets dropped because of RED.
	  • **High, TCP**—Number of high-loss priority TCP packets dropped because of RED.
	
	• (MX Series routers with enhanced DPCs, and T Series routers with enhanced FPCs only) The output classifies dropped packets into the following categories:
	  • **Low**—Number of low-loss priority packets dropped because of RED.
	  • **Medium-low**—Number of medium-low loss priority packets dropped because of RED.
	  • **Medium-high**—Number of medium-high loss priority packets dropped because of RED.
	  • **High**—Number of high-loss priority packets dropped because of RED.
	
	NOTE: Due to accounting space limitations on certain Type 3 FPCs (which are supported in M320 and T640 routers), this field does not always display the correct value for queue 6 or queue 7 for interfaces on 10-port 1-Gigabit Ethernet PICs.

Table 151: show interfaces queue Output Fields (continued)

Field Name	Field Description
RED-dropped bytes	Number of bytes dropped because of RED. The byte counts vary by interface hardware. For more information, see Table 152 on page 1359. • (M Series and T Series routers only) On M320 and M120 routers and the T Series routers, only the total number of dropped bytes is displayed. On all other M Series routers, the output classifies dropped bytes into the following categories: ◦ **Low, non-TCP**—Number of low-loss priority non-TCP bytes dropped because of RED. ◦ **Low, TCP**—Number of low-loss priority TCP bytes dropped because of RED. ◦ **High, non-TCP**—Number of high-loss priority non-TCP bytes dropped because of RED. ◦ **High, TCP**—Number of high-loss priority TCP bytes dropped because of RED. NOTE: Due to accounting space limitations on certain Type 3 FPCs (which are supported in M320 and T640 routers), this field does not always display the correct value for queue 6 or queue 7 for interfaces on 10-port 1-Gigabit Ethernet PICs.
Queue-depth bytes	Displays queue-depth average, current, peak, and maximum values for RTP queues. Because queue-depth values cannot be aggregated, displays the values for RTP queues regardless of whether **aggregate**, **remaining-traffic**, or neither option is selected.
Queue-depth bytes	Displays queue-depth average, current, peak, and maximum values for RTP queues. Because queue-depth values cannot be aggregated, displays the values for RTP queues regardless of whether **aggregate**, **remaining-traffic**, or neither option is selected.
Last-packet enqueued	Starting with Junos OS Release 16.1, **Last-packet enqueued** output field is introduced. If **packet-timestamp** is enabled for an FPC, shows the day, date, time, and year in the format *day-of-the-week month day-date hh:mm:ss yyyy* when a packet was enqueued in the CoS queue. When the timestamp is aggregated across all active Packet Forwarding Engines, the latest timestamp for each CoS queue is reported.

Byte counts vary by interface hardware. Table 152 on page 1359 shows how the byte counts on the outbound interfaces vary depending on the interface hardware. Table 152 on page 1359 is based on the assumption that outbound interfaces are sending IP traffic with 478 bytes per packet.

Table 152: Byte Count by Interface Hardware

Interface Hardware	Output Level	Byte Count Includes	Comments
Gigabit Ethernet IQ and IQE PICs	Interface	Queued: 490 bytes per packet, representing 478 bytes of Layer 3 packet + 12 bytes Transmitted: 490 bytes per packet, representing 478 bytes of Layer 3 packet + 12 bytes RED dropped: 496 bytes per packet representing 478 bytes of Layer 3 packet + 18 bytes	The 12 additional bytes include 6 bytes for the destination MAC address + 4 bytes for the VLAN + 2 bytes for the Ethernet type. For RED dropped, 6 bytes are added for the source MAC address.
	Packet forwarding component	Queued: 478 bytes per packet, representing 478 bytes of Layer 3 packet Transmitted: 478 bytes per packet, representing 478 bytes of Layer 3 packet	–
Non-IQ PIC	Interface	T Series, TX Series, T1600, and MX Series routers: • Queued: 478 bytes of Layer 3 packet. • Transmitted: 478 bytes of Layer 3 packet. T4000 routers with Type 5 FPCs : • Queued: 478 bytes of Layer 3 packet + the full Layer 2 overhead including 4 bytes CRC + the full Layer 1 overhead 8 bytes preamble + 12 bytes Inter frame Gap. • Transmitted: 478 bytes of Layer 3 packet + the full Layer 2 overhead including 4 bytes CRC + the full Layer 1 overhead 8 bytes preamble + 12 bytes Interframe Gap. M Series routers: • Queued: 478 bytes of Layer 3 packet. • Transmitted: 478 bytes of Layer 3 packet + the full Layer 2 overhead. PTX Series Packet Transport Routers: • Queued: The sum of the transmitted bytes and the RED dropped bytes. • Transmitted: Full Layer 2 overhead (including all L2 encapsulation and CRC) + 12 inter-packet gap + 8 for the preamble. • RED dropped: Full Layer 2 overhead (including all L2 encapsulation and CRC) + 12 inter-packet gap + 8 for the preamble (does not include the VLAN header or MPLS pushed bytes).	The Layer 2 overhead is 14 bytes for non-VLAN traffic and 18 bytes for VLAN traffic.

Table 152: Byte Count by Interface Hardware (continued)

Interface Hardware	Output Level	Byte Count Includes	Comments
IQ and IQE PICs with a SONET/SDH interface	Interface	Queued: 482 bytes per packet, representing 478 bytes of Layer 3 packet + 4 bytes Transmitted: 482 bytes per packet, representing 478 bytes of Layer 3 packet + 4 bytes RED dropped: 482 bytes per packet, representing 478 bytes of Layer 3 packet + 4 bytes	The additional 4 bytes are for the Layer 2 Point-to-Point Protocol (PPP) header.
	Packet forwarding component	Queued: 478 bytes per packet, representing 478 bytes of Layer 3 packet Transmitted: 486 bytes per packet, representing 478 bytes of Layer 3 packet + 8 bytes	For transmitted packets, the additional 8 bytes includes 4 bytes for the PPP header and 4 bytes for a cookie.
Non-IQ PIC with a SONET/SDH interface	Interface	T Series, TX Series, T1600, and MX Series routers: • Queued: 478 bytes of Layer 3 packet. • Transmitted: 478 bytes of Layer 3 packet. M Series routers: • Queued: 478 bytes of Layer 3 packet. • Transmitted: 483 bytes per packet, representing 478 bytes of Layer 3 packet + 5 bytes • RED dropped: 478 bytes per packet, representing 478 bytes of Layer 3 packet	For transmitted packets, the additional 5 bytes includes 4 bytes for the PPP header and 1 byte for the packet loss priority (PLP).
Interfaces configured with Frame Relay Encapsulation	Interface	The default Frame Relay overhead is 7 bytes. If you configure the Frame Check Sequence (FCS) to 4 bytes, then the overhead increases to 10 bytes.	
1-port 10-Gigabit Ethernet IQ2 and IQ2–E PICs 4-port 1G IQ2 and IQ2-E PICs 8-port 1G IQ2 and IQ2-E PICs	Interface	Queued: 478 bytes of Layer 3 packet + the full Layer 2 overhead including CRC. Transmitted: 478 bytes of Layer 3 packet + the full Layer 2 overhead including CRC.	The Layer 2 overhead is 18 bytes for non-VLAN traffic and 22 bytes for VLAN traffic.
	Packet forwarding component	Queued: 478 bytes of Layer 3 packet. Transmitted: 478 bytes of Layer 3 packet.	–

Sample Output

show interfaces queue (Rate-Limited Interface on a Gigabit Ethernet MIC in an MPC)

The following example shows queue information for the rate-limited interface ge-4/2/0 on a Gigabit Ethernet MIC in an MPC. For rate-limited queues for interfaces hosted on MICs or MPCs, rate-limit packet drops occur prior to packet output queuing. In the

command output, the nonzero statistics displayed in the **RL-dropped packets** and **RL-dropped bytes** fields quantify the traffic dropped to rate-limit queue 0 output to 10 percent of 1 gigabyte (100 megabits) per second. Because the RL-dropped traffic is not included in the **Queued** statistics, the statistics displayed for queued traffic are the same as the statistics for transmitted traffic.

```
user@host> show interfaces queue ge-4/2/0
Physical interface: ge-4/2/0, Enabled, Physical link is Up
  Interface index: 203, SNMP ifIndex: 1054
Forwarding classes: 16 supported, 4 in use
Egress queues: 8 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :            131300649                141751 pps
    Bytes                :          11287964840              99793248 bps
  Transmitted:
    Packets              :            131300649                141751 pps
    Bytes                :          11287964840              99793248 bps
    Tail-dropped packets :                    0                    0 pps
    RL-dropped packets   :            205050862                602295 pps
    RL-dropped bytes     :          13595326612              327648832 bps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
```

show interfaces queue (Aggregated Ethernet on a T320 Router)

The following example shows that the aggregated Ethernet interface, **ae1**, has traffic on queues **af1** and **af12**:

```
user@host> show interfaces queue ae1
Physical interface: ae1, Enabled, Physical link is Up
Interface index: 158, SNMP ifIndex: 33 Forwarding classes: 8 supported, 8 in use
Output queues: 8 supported, 8 in use
Queue: 0, Forwarding classes: be
  Queued:
    Packets              :                    5                    0 pps
    Bytes                :                  242                    0 bps
  Transmitted:
    Packets              :                    5                    0 pps
    Bytes                :                  242                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 1, Forwarding classes: af1
  Queued:
    Packets              :             42603765                595484 pps
```

```
    Bytes               :            5453281920             609776496 bps
  Transmitted:
    Packets             :              42603765                595484 pps
    Bytes               :            5453281920             609776496 bps
    Tail-dropped packets :                    0                     0 pps
    RED-dropped packets  :                    0                     0 pps
    RED-dropped bytes    :                    0                     0 bps
Queue: 2, Forwarding classes: ef1
  Queued:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
  Transmitted:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
    Tail-dropped packets :                    0                     0 pps
    RED-dropped packets  :                    0                     0 pps
    RED-dropped bytes    :                    0                     0 bps
Queue: 3, Forwarding classes: nc
  Queued:
    Packets             :                   45                     0 pps
    Bytes               :                 3930                     0 bps
  Transmitted:
    Packets             :                   45                     0 pps
    Bytes               :                 3930                     0 bps
    Tail-dropped packets :                    0                     0 pps
    RED-dropped packets  :                    0                     0 pps
    RED-dropped bytes    :                    0                     0 bps
Queue: 4, Forwarding classes: af11
  Queued:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
  Transmitted:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
    Tail-dropped packets :                    0                     0 pps
    RED-dropped packets  :                    0                     0 pps
    RED-dropped bytes    :                    0                     0 bps
Queue: 5, Forwarding classes: ef11
  Queued:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
  Transmitted:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
    Tail-dropped packets :                    0                     0 pps
    RED-dropped packets  :                    0                     0 pps
    RED-dropped bytes    :                    0                     0 bps
Queue: 6, Forwarding classes: af12
  Queued:
    Packets             :              31296413                437436 pps
    Bytes               :            4005940864             447935200 bps
  Transmitted:
    Packets             :              31296413                437436 pps
    Bytes               :            4005940864             447935200 bps
    Tail-dropped packets :                    0                     0 pps
    RED-dropped packets  :                    0                     0 pps
    RED-dropped bytes    :                    0                     0 bps
Queue: 7, Forwarding classes: nc2
  Queued:
    Packets             :                    0                     0 pps
    Bytes               :                    0                     0 bps
```

```
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
```

show interfaces queue (Gigabit Ethernet on a T640 Router)

```
user@host> show interfaces queue
Physical interface: ge-7/0/1, Enabled, Physical link is Up
   Interface index: 150, SNMP ifIndex: 42
 Forwarding classes: 8 supported, 8 in use
 Output queues: 8 supported, 8 in use
 Queue: 0, Forwarding classes: be
   Queued:
     Packets              :                   13                    0 pps
     Bytes                :                  622                    0 bps
   Transmitted:
     Packets              :                   13                    0 pps
     Bytes                :                  622                    0 bps
     Tail-dropped packets :                    0                    0 pps
     RED-dropped packets  :                    0                    0 pps
     RED-dropped bytes    :                    0                    0 bps
 Queue: 1, Forwarding classes: af1
   Queued:
     Packets              :           1725947945               372178 pps
     Bytes                :         220921336960            381110432 bps
   Transmitted:
     Packets              :           1725947945               372178 pps
     Bytes                :         220921336960            381110432 bps
     Tail-dropped packets :                    0                    0 pps
     RED-dropped packets  :                    0                    0 pps
     RED-dropped bytes    :                    0                    0 bps
 Queue: 2, Forwarding classes: ef1
   Queued:
     Packets              :                    0                    0 pps
     Bytes                :                    0                    0 bps
   Transmitted:
     Packets              :                    0                    0 pps
     Bytes                :                    0                    0 bps
     Tail-dropped packets :                    0                    0 pps
     RED-dropped packets  :                    0                    0 pps
     RED-dropped bytes    :                    0                    0 bps
 Queue: 3, Forwarding classes: nc
   Queued:
     Packets              :                  571                    0 pps
     Bytes                :                49318                  336 bps
   Transmitted:
     Packets              :                  571                    0 pps
     Bytes                :                49318                  336 bps
     Tail-dropped packets :                    0                    0 pps
     RED-dropped packets  :                    0                    0 pps
     RED-dropped bytes    :                    0                    0 bps
```

show interfaces queue aggregate (Gigabit Ethernet Enhanced DPC)

```
user@host> show interfaces queue ge-2/2/9 aggregate
```

```
Physical interface: ge-2/2/9, Enabled, Physical link is Up
  Interface index: 238, SNMP ifIndex: 71
Forwarding classes: 16 supported, 4 in use
Ingress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :             148450735               947295 pps
    Bytes                :            8016344944            409228848 bps
  Transmitted:
    Packets              :              76397439               487512 pps
    Bytes                :            4125461868            210602376 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :              72053285               459783 pps
     Low                 :              72053285               459783 pps
     Medium-low          :                     0                    0 pps
     Medium-high         :                     0                    0 pps
     High                :                     0                    0 pps
    RED-dropped bytes    :            3890877444            198626472 bps
     Low                 :            3890877444            198626472 bps
     Medium-low          :                     0                    0 bps
     Medium-high         :                     0                    0 bps
     High                :                     0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                     0                    0 pps
    Bytes                :                     0                    0 bps
  Transmitted:
    Packets              :                     0                    0 pps
    Bytes                :                     0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                    0 pps
     Low                 :                     0                    0 pps
     Medium-low          :                     0                    0 pps
     Medium-high         :                     0                    0 pps
     High                :                     0                    0 pps
    RED-dropped bytes    :                     0                    0 bps
     Low                 :                     0                    0 bps
     Medium-low          :                     0                    0 bps
     Medium-high         :                     0                    0 bps
     High                :                     0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :             410278257               473940 pps
    Bytes                :           22156199518            204742296 bps
  Transmitted:
    Packets              :               4850003                 4033 pps
    Bytes                :             261900162              1742256 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :             405425693               469907 pps
     Low                 :             405425693               469907 pps
     Medium-low          :                     0                    0 pps
     Medium-high         :                     0                    0 pps
     High                :                     0                    0 pps
    RED-dropped bytes    :           21892988124            203000040 bps
     Low                 :           21892988124            203000040 bps
     Medium-low          :                     0                    0 bps
     Medium-high         :                     0                    0 bps
     High                :                     0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                     0                    0 pps
```

```
    Bytes               :                     0                    0 bps
  Transmitted:
    Packets             :                     0                    0 pps
    Bytes               :                     0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                    0 pps
     Low                :                     0                    0 pps
     Medium-low         :                     0                    0 pps
     Medium-high        :                     0                    0 pps
     High               :                     0                    0 pps
    RED-dropped bytes    :                     0                    0 bps
     Low                :                     0                    0 bps
     Medium-low         :                     0                    0 bps
     Medium-high        :                     0                    0 bps
     High               :                     0                    0 bps
Forwarding classes: 16 supported, 4 in use
Egress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets             :              76605230               485376 pps
    Bytes               :            5209211400            264044560 bps
  Transmitted:
    Packets             :              76444631               484336 pps
    Bytes               :            5198235612            263478800 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                160475                 1040 pps
     Low                :                160475                 1040 pps
     Medium-low         :                     0                    0 pps
     Medium-high        :                     0                    0 pps
     High               :                     0                    0 pps
    RED-dropped bytes    :              10912300               565760 bps
     Low                :              10912300               565760 bps
     Medium-low         :                     0                    0 bps
     Medium-high        :                     0                    0 bps
     High               :                     0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets             :                     0                    0 pps
    Bytes               :                     0                    0 bps
  Transmitted:
    Packets             :                     0                    0 pps
    Bytes               :                     0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                    0 pps
     Low                :                     0                    0 pps
     Medium-low         :                     0                    0 pps
     Medium-high        :                     0                    0 pps
     High               :                     0                    0 pps
    RED-dropped bytes    :                     0                    0 bps
     Low                :                     0                    0 bps
     Medium-low         :                     0                    0 bps
     Medium-high        :                     0                    0 bps
     High               :                     0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets             :               4836136                 3912 pps
    Bytes               :             333402032              2139056 bps
  Transmitted:
    Packets             :               3600866                 1459 pps
    Bytes               :             244858888               793696 bps
    Tail-dropped packets : Not Available
```

```
    RED-dropped packets  :               1225034                2450 pps
     Low                 :               1225034                2450 pps
     Medium-low          :                     0                   0 pps
     Medium-high         :                     0                   0 pps
     High                :                     0                   0 pps
    RED-dropped bytes    :              83302312             1333072 bps
     Low                 :              83302312             1333072 bps
     Medium-low          :                     0                   0 bps
     Medium-high         :                     0                   0 bps
     High                :                     0                   0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                     0                   0 pps
    Bytes                :                     0                   0 bps
  Transmitted:
    Packets              :                     0                   0 pps
    Bytes                :                     0                   0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                   0 pps
     Low                 :                     0                   0 pps
     Medium-low          :                     0                   0 pps
     Medium-high         :                     0                   0 pps
     High                :                     0                   0 pps
    RED-dropped bytes    :                     0                   0 bps
     Low                 :                     0                   0 bps
     Medium-low          :                     0                   0 bps
     Medium-high         :                     0                   0 bps
     High                :                     0                   0 bps

Packet Forwarding Engine Chassis Queues:
Queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :              77059796              486384 pps
    Bytes                :            3544750624           178989576 bps
  Transmitted:
    Packets              :              77059797              486381 pps
    Bytes                :            3544750670           178988248 bps
    Tail-dropped packets :                     0                   0 pps
    RED-dropped packets  :                     0                   0 pps
     Low                 :                     0                   0 pps
     Medium-low          :                     0                   0 pps
     Medium-high         :                     0                   0 pps
     High                :                     0                   0 pps
    RED-dropped bytes    :                     0                   0 bps
     Low                 :                     0                   0 bps
     Medium-low          :                     0                   0 bps
     Medium-high         :                     0                   0 bps
     High                :                     0                   0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                     0                   0 pps
    Bytes                :                     0                   0 bps
  Transmitted:
    Packets              :                     0                   0 pps
    Bytes                :                     0                   0 bps
    Tail-dropped packets :                     0                   0 pps
    RED-dropped packets  :                     0                   0 pps
     Low                 :                     0                   0 pps
     Medium-low          :                     0                   0 pps
     Medium-high         :                     0                   0 pps
```

```
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :              4846580                 3934 pps
    Bytes                :            222942680              1447768 bps
  Transmitted:
    Packets              :              4846580                 3934 pps
    Bytes                :            222942680              1447768 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
```

show interfaces queue (Gigabit Ethernet IQ2 PIC)

```
user@host> show interfaces queue ge-7/1/3
Physical interface: ge-7/1/3, Enabled, Physical link is Up
  Interface index: 170, SNMP ifIndex: 70 Forwarding classes: 16 supported, 4 in
use Ingress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :            418390039                   10 pps
    Bytes                :          38910269752                 7440 bps
  Transmitted:
    Packets              :            418390039                   10 pps
    Bytes                :          38910269752                 7440 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
```

```
  Queued:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
  Transmitted:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
  Transmitted:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                  7055                     1 pps
    Bytes                :                451552                   512 bps
  Transmitted:
    Packets              :                  7055                     1 pps
    Bytes                :                451552                   512 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Forwarding classes: 16 supported, 4 in use Egress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :                  1031                     0 pps
    Bytes                :                143292                     0 bps
  Transmitted:
    Packets              :                  1031                     0 pps
    Bytes                :                143292                     0 bps
    Tail-dropped packets : Not Available
    RL-dropped packets   :                     0                     0 pps
    RL-dropped bytes     :                     0                     0 bps
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
  Transmitted:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
    Tail-dropped packets : Not Available
    RL-dropped packets   :                     0                     0 pps
    RL-dropped bytes     :                     0                     0 bps
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
  Transmitted:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
```

```
    Tail-dropped packets : Not Available
    RL-dropped packets   :                      0                      0 pps
    RL-dropped bytes     :                      0                      0 bps
    RED-dropped packets  :                      0                      0 pps
    RED-dropped bytes    :                      0                      0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                  77009                     11 pps
    Bytes                :                6894286                   7888 bps
  Transmitted:
    Packets              :                  77009                     11 pps
    Bytes                :                6894286                   7888 bps
    Tail-dropped packets : Not Available
    RL-dropped packets   :                      0                      0 pps
    RL-dropped bytes     :                      0                      0 bps
    RED-dropped packets  :                      0                      0 pps
    RED-dropped bytes    :                      0                      0 bps

Packet Forwarding Engine Chassis Queues:
Queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :                   1031                      0 pps
    Bytes                :                 147328                      0 bps
  Transmitted:
    Packets              :                   1031                      0 pps
    Bytes                :                 147328                      0 bps
    Tail-dropped packets :                      0                      0 pps
    RED-dropped packets  :                      0                      0 pps
     Low, non-TCP        :                      0                      0 pps
     Low, TCP            :                      0                      0 pps
     High, non-TCP       :                      0                      0 pps
     High, TCP           :                      0                      0 pps
    RED-dropped bytes    :                      0                      0 bps
     Low, non-TCP        :                      0                      0 bps
     Low, TCP            :                      0                      0 bps
     High, non-TCP       :                      0                      0 bps
     High, TCP           :                      0                      0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                      0                      0 pps
    Bytes                :                      0                      0 bps
  Transmitted:
    Packets              :                      0                      0 pps
    Bytes                :                      0                      0 bps
    Tail-dropped packets :                      0                      0 pps
    RED-dropped packets  :                      0                      0 pps
     Low, non-TCP        :                      0                      0 pps
     Low, TCP            :                      0                      0 pps
     High, non-TCP       :                      0                      0 pps
     High, TCP           :                      0                      0 pps
    RED-dropped bytes    :                      0                      0 bps
     Low, non-TCP        :                      0                      0 bps
     Low, TCP            :                      0                      0 bps
     High, non-TCP       :                      0                      0 bps
     High, TCP           :                      0                      0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                      0                      0 pps
    Bytes                :                      0                      0 bps
  Transmitted:
```

```
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low, non-TCP        :                    0                    0 pps
     Low, TCP            :                    0                    0 pps
     High, non-TCP       :                    0                    0 pps
     High, TCP           :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low, non-TCP        :                    0                    0 bps
     Low, TCP            :                    0                    0 bps
     High, non-TCP       :                    0                    0 bps
     High, TCP           :                    0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                94386                   12 pps
    Bytes                :             13756799                 9568 bps
  Transmitted:
    Packets              :                94386                   12 pps
    Bytes                :             13756799                 9568 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low, non-TCP        :                    0                    0 pps
     Low, TCP            :                    0                    0 pps
     High, non-TCP       :                    0                    0 pps
     High, TCP           :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low, non-TCP        :                    0                    0 bps
     Low, TCP            :                    0                    0 bps
     High, non-TCP       :                    0                    0 bps
     High, TCP           :                    0                    0 bps
```

show interfaces queue both-ingress-egress (Gigabit Ethernet IQ2 PIC)

```
user@host> show interfaces queue ge-6/2/0 both-ingress-egress
Physical interface: ge-6/2/0, Enabled, Physical link is Up
  Interface index: 175, SNMP ifIndex: 121
Forwarding classes: 8 supported, 4 in use
Ingress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                  254                    0 pps
    Bytes                :                16274                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
```

```
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Forwarding classes: 8 supported, 4 in use
Egress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       3                       0 pps
    Bytes                :                     126                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
```

```
Packet Forwarding Engine Chassis Queues:
Queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :              80564692                    0 pps
    Bytes                :            3383717100                    0 bps
  Transmitted:
    Packets              :              80564692                    0 pps
    Bytes                :            3383717100                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :              80564685                    0 pps
    Bytes                :            3383716770                    0 bps
  Transmitted:
    Packets              :              80564685                    0 pps
    Bytes                :            3383716770                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                 9397                    0 pps
    Bytes                :              3809052                  232 bps
  Transmitted:
    Packets              :                 9397                    0 pps
    Bytes                :              3809052                  232 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
```

show interfaces queue ingress (Gigabit Ethernet IQ2 PIC)

```
user@host> show interfaces queue ge-6/2/0 ingress
Physical interface: ge-6/2/0, Enabled, Physical link is Up
  Interface index: 175, SNMP ifIndex: 121
Forwarding classes: 8 supported, 4 in use
Ingress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                  288                    0 pps
    Bytes                :                18450                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
```

```
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
```

show interfaces queue egress (Gigabit Ethernet IQ2 PIC)

```
user@host> show interfaces queue ge-6/2/0 egress
Physical interface: ge-6/2/0, Enabled, Physical link is Up
  Interface index: 175, SNMP ifIndex: 121
Forwarding classes: 8 supported, 4 in use
Egress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    3                    0 pps
    Bytes                :                  126                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              : Not Available
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
```

```
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              : Not Available
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Packet Forwarding Engine Chassis Queues:
Queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :                80564692                       0 pps
    Bytes                :              3383717100                       0 bps
  Transmitted:
    Packets              :                80564692                       0 pps
    Bytes                :              3383717100                       0 bps
    Tail-dropped packets :                       0                       0 pps
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                80564685                       0 pps
    Bytes                :              3383716770                       0 bps
  Transmitted:
    Packets              :                80564685                       0 pps
    Bytes                :              3383716770                       0 bps
    Tail-dropped packets :                       0                       0 pps
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
  Transmitted:
    Packets              :                       0                       0 pps
    Bytes                :                       0                       0 bps
    Tail-dropped packets :                       0                       0 pps
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                    9538                       0 pps
    Bytes                :                 3819840                       0 bps
  Transmitted:
    Packets              :                    9538                       0 pps
    Bytes                :                 3819840                       0 bps
    Tail-dropped packets :                       0                       0 pps
    RED-dropped packets  :                       0                       0 pps
    RED-dropped bytes    :                       0                       0 bps
```

show interfaces queue remaining-traffic (Gigabit Ethernet Enhanced DPC)

```
user@host> show interfaces queue ge-2/2/9 remaining-traffic
Physical interface: ge-2/2/9, Enabled, Physical link is Up
  Interface index: 238, SNMP ifIndex: 71
Forwarding classes: 16 supported, 4 in use
Ingress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :            110208969               472875 pps
    Bytes                :           5951284434            204282000 bps
  Transmitted:
    Packets              :            110208969               472875 pps
    Bytes                :           5951284434            204282000 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
```

```
     Medium-high         :                      0                    0 bps
     High                :                      0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                      0                    0 pps
    Bytes                :                      0                    0 bps
  Transmitted:
    Packets              :                      0                    0 pps
    Bytes                :                      0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                      0                    0 pps
     Low                 :                      0                    0 pps
     Medium-low          :                      0                    0 pps
     Medium-high         :                      0                    0 pps
     High                :                      0                    0 pps
    RED-dropped bytes    :                      0                    0 bps
     Low                 :                      0                    0 bps
     Medium-low          :                      0                    0 bps
     Medium-high         :                      0                    0 bps
     High                :                      0                    0 bps
Forwarding classes: 16 supported, 4 in use
Egress queues: 4 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets              :              109355853               471736 pps
    Bytes                :             7436199152            256627968 bps
  Transmitted:
    Packets              :              109355852               471736 pps
    Bytes                :             7436198640            256627968 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                      0                    0 pps
     Low                 :                      0                    0 pps
     Medium-low          :                      0                    0 pps
     Medium-high         :                      0                    0 pps
     High                :                      0                    0 pps
    RED-dropped bytes    :                      0                    0 bps
     Low                 :                      0                    0 bps
     Medium-low          :                      0                    0 bps
     Medium-high         :                      0                    0 bps
     High                :                      0                    0 bps
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets              :                      0                    0 pps
    Bytes                :                      0                    0 bps
  Transmitted:
    Packets              :                      0                    0 pps
    Bytes                :                      0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                      0                    0 pps
     Low                 :                      0                    0 pps
     Medium-low          :                      0                    0 pps
     Medium-high         :                      0                    0 pps
     High                :                      0                    0 pps
    RED-dropped bytes    :                      0                    0 bps
     Low                 :                      0                    0 bps
     Medium-low          :                      0                    0 bps
     Medium-high         :                      0                    0 bps
     High                :                      0                    0 bps
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets              :                      0                    0 pps
```

```
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets : Not Available
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
```

show interfaces queue (Channelized OC12 IQE Type 3 PIC in SONET Mode)

```
user@host> show interfaces queue t3-1/1/0:7
 Physical interface: t3-1/1/0:7, Enabled, Physical link is Up

   Interface index: 192, SNMP ifIndex: 1948

   Description: full T3 interface connect to 6ce13 t3-3/1/0:7 for FR testing -
Lam

 Forwarding classes: 16 supported, 9 in use

 Egress queues: 8 supported, 8 in use

 Queue: 0, Forwarding classes: DEFAULT

   Queued:

     Packets              :               214886                13449 pps

     Bytes                :              9884756              5164536 bps

   Transmitted:

     Packets              :               214886                13449 pps
```

```
    Bytes                :              9884756              5164536 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 1, Forwarding classes: REALTIME
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 2, Forwarding classes: PRIVATE
```

```
  Queued:
    Packets              :                 0                 0 pps
    Bytes                :                 0                 0 bps
  Transmitted:
    Packets              :                 0                 0 pps
    Bytes                :                 0                 0 bps
    Tail-dropped packets :                 0                 0 pps
    RED-dropped packets  :                 0                 0 pps
     Low                 :                 0                 0 pps
     Medium-low          :                 0                 0 pps
     Medium-high         :                 0                 0 pps
     High                :                 0                 0 pps
    RED-dropped bytes    :                 0                 0 bps
     Low                 :                 0                 0 bps
     Medium-low          :                 0                 0 bps
     Medium-high         :                 0                 0 bps
     High                :                 0                 0 bps
Queue: 3, Forwarding classes: CONTROL
  Queued:
    Packets              :                60                 0 pps
    Bytes                :              4560                 0 bps
  Transmitted:
    Packets              :                60                 0 pps
    Bytes                :              4560                 0 bps
    Tail-dropped packets :                 0                 0 pps
    RED-dropped packets  :                 0                 0 pps
     Low                 :                 0                 0 pps
     Medium-low          :                 0                 0 pps
     Medium-high         :                 0                 0 pps
     High                :                 0                 0 pps
```

```
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 4, Forwarding classes: CLASS_B_OUTPUT
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 5, Forwarding classes: CLASS_C_OUTPUT
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
```

```
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 6, Forwarding classes: CLASS_V_OUTPUT
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 7, Forwarding classes: CLASS_S_OUTPUT, GETS
  Queued:
```

```
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps

Packet Forwarding Engine Chassis Queues:
Queues: 8 supported, 8 in use
Queue: 0, Forwarding classes: DEFAULT
  Queued:
    Packets              :               371365                23620 pps
    Bytes                :             15597330              7936368 bps
  Transmitted:
    Packets              :               371365                23620 pps
    Bytes                :             15597330              7936368 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
```

```
      High                 :                    0                    0 pps
     RED-dropped bytes     :                    0                    0 bps
      Low                  :                    0                    0 bps
      Medium-low           :                    0                    0 bps
      Medium-high          :                    0                    0 bps
      High                 :                    0                    0 bps
Queue: 1, Forwarding classes: REALTIME
  Queued:
    Packets                :                    0                    0 pps
    Bytes                  :                    0                    0 bps
  Transmitted:
    Packets                :                    0                    0 pps
    Bytes                  :                    0                    0 bps
    Tail-dropped packets   :                    0                    0 pps
    RED-dropped packets    :                    0                    0 pps
      Low                  :                    0                    0 pps
      Medium-low           :                    0                    0 pps
      Medium-high          :                    0                    0 pps
      High                 :                    0                    0 pps
     RED-dropped bytes     :                    0                    0 bps
      Low                  :                    0                    0 bps
      Medium-low           :                    0                    0 bps
      Medium-high          :                    0                    0 bps
      High                 :                    0                    0 bps
Queue: 2, Forwarding classes: PRIVATE
  Queued:
    Packets                :                    0                    0 pps
    Bytes                  :                    0                    0 bps
  Transmitted:
    Packets                :                    0                    0 pps
```

```
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 3, Forwarding classes: CONTROL
  Queued:
    Packets              :                32843                    0 pps
    Bytes                :              2641754                   56 bps
  Transmitted:
    Packets              :                32843                    0 pps
    Bytes                :              2641754                   56 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 4, Forwarding classes: CLASS_B_OUTPUT
```

```
  Queued:
    Packets              :                 0                 0 pps
    Bytes                :                 0                 0 bps
  Transmitted:
    Packets              :                 0                 0 pps
    Bytes                :                 0                 0 bps
    Tail-dropped packets :                 0                 0 pps
    RED-dropped packets  :                 0                 0 pps
     Low                 :                 0                 0 pps
     Medium-low          :                 0                 0 pps
     Medium-high         :                 0                 0 pps
     High                :                 0                 0 pps
    RED-dropped bytes    :                 0                 0 bps
     Low                 :                 0                 0 bps
     Medium-low          :                 0                 0 bps
     Medium-high         :                 0                 0 bps
     High                :                 0                 0 bps
Queue: 5, Forwarding classes: CLASS_C_OUTPUT
  Queued:
    Packets              :                 0                 0 pps
    Bytes                :                 0                 0 bps
  Transmitted:
    Packets              :                 0                 0 pps
    Bytes                :                 0                 0 bps
    Tail-dropped packets :                 0                 0 pps
    RED-dropped packets  :                 0                 0 pps
     Low                 :                 0                 0 pps
     Medium-low          :                 0                 0 pps
     Medium-high         :                 0                 0 pps
     High                :                 0                 0 pps
```

```
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 6, Forwarding classes: CLASS_V_OUTPUT
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 7, Forwarding classes: CLASS_S_OUTPUT, GETS
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
```

```
      RED-dropped packets  :                    0                    0 pps
       Low                 :                    0                    0 pps
       Medium-low          :                    0                    0 pps
       Medium-high         :                    0                    0 pps
       High                :                    0                    0 pps
      RED-dropped bytes    :                    0                    0 bps
       Low                 :                    0                    0 bps
       Medium-low          :                    0                    0 bps
       Medium-high         :                    0                    0 bps
       High                :                    0                    0 bps
```

show interfaces queue (QFX Series)

```
user@switch> show interfaces queue xe-0/0/15
   Physical interface: xe-0/0/15, Enabled, Physical link is Up
     Interface index: 49165, SNMP ifIndex: 539
   Forwarding classes: 12 supported, 8 in use
   Egress queues: 12 supported, 8 in use
   Queue: 0, Forwarding classes: best-effort
     Queued:
       Packets              :                    0                    0 pps
       Bytes                :                    0                    0 bps
     Transmitted:
       Packets              :                    0                    0 pps
       Bytes                :                    0                    0 bps
       Tail-dropped packets : Not Available
       Total-dropped packets:                    0                    0 pps
       Total-dropped bytes  :                    0                    0 bps
   Queue: 3, Forwarding classes: fcoe
     Queued:
       Packets              :                    0                    0 pps
       Bytes                :                    0                    0 bps
     Transmitted:
       Packets              :                    0                    0 pps
       Bytes                :                    0                    0 bps
       Tail-dropped packets : Not Available
       Total-dropped packets:                    0                    0 pps
       Total-dropped bytes  :                    0                    0 bps
0 bps
   Queue: 4, Forwarding classes: no-loss
     Queued:
       Packets              :                    0                    0 pps
       Bytes                :                    0                    0 bps
     Transmitted:
       Packets              :                    0                    0 pps
       Bytes                :                    0                    0 bps
       Tail-dropped packets : Not Available
       Total-dropped packets:                    0                    0 pps
       Total-dropped bytes  :                    0                    0 bps
```

```
  Queue: 7, Forwarding classes: network-control
    Queued:
      Packets              :                     0                     0 pps
      Bytes                :                     0                     0 bps
    Transmitted:
      Packets              :                     0                     0 pps
      Bytes                :                     0                     0 bps
      Tail-dropped packets : Not Available
      Total-dropped packets:                     0                     0 pps
      Total-dropped bytes  :                     0                     0 bps
  Queue: 8, Forwarding classes: mcast
    Queued:
      Packets              :                     0                     0 pps
      Bytes                :                     0                     0 bps
    Transmitted:
      Packets              :                     0                     0 pps
      Bytes                :                     0                     0 bps
      Tail-dropped packets : Not Available
      Total-dropped packets:                     0                     0 pps
      Total-dropped bytes  :                     0                     0 bps
```

show interfaces queue l2-statistics (lsq interface)

```
user@switch> show interfaces queue lsq-2/2/0.2 l2-statistics
Logical interface lsq-2/2/0.2 (Index 69) (SNMP ifIndex 1598)
Forwarding classes: 16 supported, 4 in use
Egress queues: 8 supported, 4 in use
Burst size: 0
Queue: 0, Forwarding classes: be
  Queued:
    Packets              :                     1                     0 pps
    Bytes                :                  1001                     0 bps
  Transmitted:
    Packets              :                     5                      0 pps
    Bytes                :                  1062                     0 bps
    Tail-dropped packets :                     0                     0 pps
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 1, Forwarding classes: ef
  Queued:
    Packets              :                     1                     0 pps
    Bytes                :                  1500                     0 bps
  Transmitted:
    Packets              :                     6                      0 pps
    Bytes                :                  1573                     0 bps
    Tail-dropped packets :                     0                     0 pps
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 2, Forwarding classes: af
  Queued:
    Packets              :                     1                     0 pps
    Bytes                :                   512                     0 bps
  Transmitted:
    Packets              :                     3                      0 pps
    Bytes                :                   549                     0 bps
    Tail-dropped packets :                     0                     0 pps
    RED-dropped packets  :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
Queue: 3, Forwarding classes: nc
  Queued:
```

```
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RED-dropped packets  :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
=========
```

show interfaces queue lsq (lsq-ifd)

```
user@switch> show interfaces queue lsq-1/0/0
Logical interface lsq-1/0/0 (Index 348) (SNMP ifIndex 660)
Forwarding classes: 16 supported, 4 in use
Egress queues: 8 supported, 4 in use
Burst size: 0
Queue: 0, Forwarding classes: be
  Queued:
    Packets              :                55576                 1206 pps
    Bytes                :             29622008              5145472 bps
  Transmitted:
    Packets              :                55576                 1206 pps
    Bytes                :             29622008              5145472 bps
    Tail-dropped packets :                    0                    0 pps
    RL-dropped packets   :                    0                    0 pps
    RL-dropped bytes     :                    0                    0 bps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 1, Forwarding classes: ef
  Queued:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
  Transmitted:
    Packets              :                    0                    0 pps
    Bytes                :                    0                    0 bps
    Tail-dropped packets :                    0                    0 pps
    RL-dropped packets   :                    0                    0 pps
    RL-dropped bytes     :                    0                    0 bps
    RED-dropped packets  :                    0                    0 pps
     Low                 :                    0                    0 pps
     Medium-low          :                    0                    0 pps
     Medium-high         :                    0                    0 pps
     High                :                    0                    0 pps
    RED-dropped bytes    :                    0                    0 bps
     Low                 :                    0                    0 bps
     Medium-low          :                    0                    0 bps
     Medium-high         :                    0                    0 bps
     High                :                    0                    0 bps
Queue: 2, Forwarding classes: af
  Queued:
    Packets              :                    0                    0 pps
```

```
    Bytes               :                    0                    0 bps
  Transmitted:
    Packets             :                    0                    0 pps
    Bytes               :                    0                    0 bps
    Tail-dropped packets :                   0                    0 pps
    RL-dropped packets  :                    0                    0 pps
    RL-dropped bytes    :                    0                    0 bps
    RED-dropped packets :                    0                    0 pps
     Low                :                    0                    0 pps
     Medium-low         :                    0                    0 pps
     Medium-high        :                    0                    0 pps
     High               :                    0                    0 pps
    RED-dropped bytes   :                    0                    0 bps
     Low                :                    0                    0 bps
     Medium-low         :                    0                    0 bps
     Medium-high        :                    0                    0 bps
     High               :                    0                    0 bps
Queue: 3, Forwarding classes: nc
  Queued:
    Packets             :                22231                  482 pps
    Bytes               :             11849123              2057600 bps
  Transmitted:
    Packets             :                22231                  482 pps
    Bytes               :             11849123              2057600 bps
    Tail-dropped packets :                   0                    0 pps
    RL-dropped packets  :                    0                    0 pps
    RL-dropped bytes    :                    0                    0 bps
    RED-dropped packets :                    0                    0 pps
     Low                :                    0                    0 pps
     Medium-low         :                    0                    0 pps
     Medium-high        :                    0                    0 pps
     High               :                    0                    0 pps
    RED-dropped bytes   :                    0                    0 bps
     Low                :                    0                    0 bps
     Medium-low         :                    0                    0 bps
     Medium-high        :                    0                    0 bps
     High               :                    0                    0 bps
```

Sample Output

show interfaces queue (Aggregated Ethernet on a MX series Router)

```
user@host> show interfaces queue ae0 remaining-traffic

Physical interface: ae0    , Enabled, Physical link is Up
  Interface index: 128, SNMP ifIndex: 543
Forwarding classes: 16 supported, 4 in use
Egress queues: 8 supported, 4 in use
Queue: 0, Forwarding classes: best-effort
  Queued:
    Packets             :                   16                    0 pps
    Bytes               :                 1896                    0 bps
  Transmitted:
    Packets             :                   16                    0 pps
    Bytes               :                 1896                    0 bps
    Tail-dropped packets :                   0                    0 pps
    RL-dropped packets  :                    0                    0 pps
    RL-dropped bytes    :                    0                    0 bps
    RED-dropped packets :                    0                    0 pps
     Low                :                    0                    0 pps
     Medium-low         :                    0                    0 pps
```

```
      Medium-high         :                    0                    0 pps
      High                :                    0                    0 pps
    RED-dropped bytes     :                    0                    0 bps
      Low                 :                    0                    0 bps
      Medium-low          :                    0                    0 bps
      Medium-high         :                    0                    0 bps
      High                :                    0                    0 bps
  Queue-depth bytes       :
    Average               :                    0
    Current               :                    0
    Peak                  :                    0
    Maximum               :            119013376
Queue: 1, Forwarding classes: expedited-forwarding
  Queued:
    Packets               :                    0                    0 pps
    Bytes                 :                    0                    0 bps
  Transmitted:
    Packets               :                    0                    0 pps
    Bytes                 :                    0                    0 bps
    Tail-dropped packets  :                    0                    0 pps
    RL-dropped packets    :                    0                    0 pps
    RL-dropped bytes      :                    0                    0 bps
    RED-dropped packets   :                    0                    0 pps
      Low                 :                    0                    0 pps
      Medium-low          :                    0                    0 pps
      Medium-high         :                    0                    0 pps
      High                :                    0                    0 pps
    RED-dropped bytes     :                    0                    0 bps
      Low                 :                    0                    0 bps
      Medium-low          :                    0                    0 bps
      Medium-high         :                    0                    0 bps
      High                :                    0                    0 bps
  Queue-depth bytes       :
    Average               :                    0
    Current               :                    0
    Peak                  :                    0
    Maximum               :                32768
Queue: 2, Forwarding classes: assured-forwarding
  Queued:
    Packets               :                    0                    0 pps
    Bytes                 :                    0                    0 bps
  Transmitted:
    Packets               :                    0                    0 pps
    Bytes                 :                    0                    0 bps
    Tail-dropped packets  :                    0                    0 pps
    RL-dropped packets    :                    0                    0 pps
    RL-dropped bytes      :                    0                    0 bps
    RED-dropped packets   :                    0                    0 pps
      Low                 :                    0                    0 pps
      Medium-low          :                    0                    0 pps
      Medium-high         :                    0                    0 pps
      High                :                    0                    0 pps
    RED-dropped bytes     :                    0                    0 bps
      Low                 :                    0                    0 bps
      Medium-low          :                    0                    0 bps
      Medium-high         :                    0                    0 bps
      High                :                    0                    0 bps
  Queue-depth bytes       :
    Average               :                    0
    Current               :                    0
    Peak                  :                    0
```

```
    Maximum              :                 32768
Queue: 3, Forwarding classes: network-control
  Queued:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
  Transmitted:
    Packets              :                     0                     0 pps
    Bytes                :                     0                     0 bps
    Tail-dropped packets :                     0                     0 pps
    RL-dropped packets   :                     0                     0 pps
    RL-dropped bytes     :                     0                     0 bps
    RED-dropped packets  :                     0                     0 pps
     Low                 :                     0                     0 pps
     Medium-low          :                     0                     0 pps
     Medium-high         :                     0                     0 pps
     High                :                     0                     0 pps
    RED-dropped bytes    :                     0                     0 bps
     Low                 :                     0                     0 bps
     Medium-low          :                     0                     0 bps
     Medium-high         :                     0                     0 bps
     High                :                     0                     0 bps
  Queue-depth bytes      :
    Average              :                     0
    Current              :                     0
    Peak                 :                     0
    Maximum              :               6258688
```

show interfaces swfabx

Syntax show interfaces (swfab0 | swfab1)

Release Information Command introduced in Junos OS Release 11.1.

Description Display the configured interfaces for each swfab interface. The swfab interface can contain one or more members because it is an aggregated interface.

Required Privilege Level view

Related Documentation
- clear interfaces statistics swfabx on page 1175

List of Sample Output
show interfaces swfab0 on page 1393
show interfaces swfab1 on page 1393

Output Fields Table 153 on page 1393 lists the output fields for the **show interfaces <swfab0 | swfab1>** command. Output fields are listed in the approximate order in which they appear.

Table 153: show interfaces <swfab0 | swfab1> Output Fields

Field Name	Field Description
fabric-options	The fabric-options hierarchy is configured to be in sync with the fab interfaces.
member-interfaces	Interfaces specified under member-interfaces are single aggregate interfaces. This interface carries internode switching traffic.

Sample Output

show interfaces swfab0

```
user@host# show interfaces swfab0
fabric-options {
       member-interfaces {
           ge-0/0/9;
           ge-0/0/10;
       }
  }
```

show interfaces swfab1

```
user@host# show interfaces swfab1
fabric-options {
       member-interfaces {
           ge-7/0/9;
           ge-7/0/10;
```

```
            }
        }
```

show mac-rewrite interface

Syntax

show mac-rewrite interface
<brief | detail>
<interface-name>

Release Information

Command introduced in Junos OS Release 9.1.
Command introduced in Junos OS Release 14.1X53-D10 for EX4300 switches.
Command introducced in Junos OS Release 15.1X53-D55 for EX2300 and EX3400 switches.
Command introduced in Junos OS Release 17.4R1 for EX4600 switches.

Description

Display Layer 2 protocol tunneling (L2PT) information.

Options

brief | detail—(Optional) Display the specified level of output.

interface *interface-name*—(Optional) Display L2PT information for the specified interface.

Required Privilege Level

view

Related Documentation

- layer2-control on page 1016
- mac-rewrite on page 1031
- protocol on page 1084
- *Layer 2 Protocol Tunneling Through a Network*
- *Layer 2 Protocol Tunnel Configuration Guidelines*
- *Configuring Layer 2 Protocol Tunneling*
- Understanding Layer 2 Protocol Tunneling on EX Series Switches on page 389
- Configuring Layer 2 Protocol Tunneling on EX Series Switches with ELS Support (CLI Procedure) on page 398

List of Sample Output

show mac-rewrite interface on page 1396
show mac-rewrite interface (EX Series Switches) on page 1396

Output Fields

Table 154 on page 1395 lists the output fields for the **show mac-rewrite interface** command. Output fields are listed in the approximate order in which they appear.

Table 154: show mac-rewrite interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface on which L2PT is configured.	**brief detail**

Table 154: show mac-rewrite interface Output Fields (continued)

Field Name	Field Description	Level of Output
Protocols	Layer 2 protocols being tunneled on this interface. All devices that support L2PT can tunnel the following protocols: Cisco Discovery Protocol (CDP), Spanning Tree Protocol (STP), or VLAN Trunk Protocol (VTP). The following Layer 2 protocols can also be tunneled on some devices that support L2PT: E-LMI, GVRP, IEEE 802.1X, IEEE 802.3AH, LACP, LLDP, MMRP, MVRP, PVSTP+, UDLD, or VSTP. See protocol for more information on the supported protocols for tunneling on different devices.	**brief detail**

Sample Output

show mac-rewrite interface

```
user@host> show mac-rewrite interface
Interface       Protocols

 ge-1/0/5       STP VTP CDP PVSTP+
```

show mac-rewrite interface (EX Series Switches)

```
user@switch> show mac-rewrite interface
Interface       Protocols

 ge-0/0/1       802.3AH LLDP STP
```

show mvrp

Syntax show mvrp

Release Information
Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 10.1 for MX Series routers.
Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description Display Multiple VLAN Registration Protocol (MVRP) configuration information.

Required Privilege Level view

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Verifying That MVRP Is Working Correctly on Switches on page 548
- show mvrp statistics on page 1408
- show mvrp applicant-state on page 1400
- show mvrp dynamic-vlan-memberships on page 1402
- show mvrp interface on page 1404
- show mvrp registration-state on page 1406
- show mvrp statistics on page 1408
- show mvrp applicant-state on page 1400
- show mvrp dynamic-vlan-memberships on page 1402
- show mvrp interface on page 1404
- show mvrp registration-state on page 1406

List of Sample Output
show mvrp (EX Series switches and MX Series routers) on page 1398
show mvrp (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320) on page 1398
show mvrp (EX Series switches) on page 1399

Output Fields Table 155 on page 1397 lists the output fields for the **show mvrp** command. Output fields are listed in the approximate order in which they appear.

Table 155: show mvrp Output Fields

Field Name	Field Description
MVRP dynamic VLAN creation	Displays whether global MVRP dynamic VLAN creation is **Enabled** or **Disabled**.

Table 155: show mvrp Output Fields (continued)

Field Name	Field Description
Global MVRP configuration	Displays global MVRP information: • **MVRP status**—Displays whether MVRP is **Enabled** or **Disabled**. • **MVRP dynamic vlan creation**—Displays whether global MVRP dynamic VLAN creation is **Dnabled** or **Disabled**.
MVRP BPDU MAC address	Displays the multicast media access control (MAC) address for MVRP. If configured, the provider MVRP multicast MAC address is used; otherwise, the customer MVRP multicast MAC address is used.
MVRP timers (ms)	Displays MVRP timer information: • **Interface**—The interface on which MVRP is configured. • **Join**—The maximum number of milliseconds the interfaces must wait before sending VLAN advertisements. • **Leave**—The number of milliseconds an interface must wait after receiving a Leave message to remove the interface from the VLAN specified in the message. • **LeaveAll**— The interval at which LeaveAll messages are sent on interfaces. LeaveAll messages maintain current MVRP VLAN membership information in the network.
Interface based configuration	Displays interface-specific MVRP information: • **Interface**—The interface on which MVRP is configured. • **Status**—Displays whether MVRP is **Enabled** or **Disabled**. • **Registration**—Displays whether registration for the interface is **Forbidden** or **Normal**. • **Dynamic VLAN Creation**—Displays whether interface dynamic VLAN creation is **Enabled** or **Disabled**.

Sample Output

show mvrp (EX Series switches and MX Series routers)

```
user@host> show mvrp
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (01-80-C2-00-00-21)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  ge-11/2/8        200     800     10000
  ge-11/0/9        200     800     10000
  ge-11/3/0        200     800     10000
```

Sample Output

show mvrp (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

```
user@host> show mvrp
MVRP configuration for routing instance 'default-switch'
MVRP dynamic VLAN creation : Enabled
MVRP BPDU MAC address      : Customer bridge group (00-00-5E-00-53-00)
MVRP timers (ms)
  Interface       Join   Leave  LeaveAll
  ge-0/0/1        200     800     60
```

Sample Output

show mvrp (EX Series switches)

```
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   Normal    Enabled
   xe-0/1/1.0     Enabled    Normal         Enabled
```

show mvrp applicant-state

Syntax show mvrp applicant-state

Release Information Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description For MX Series routers, EX Series switches, SRX1500, SRX300, SRX550M, SRX345, SRX340, and SRX320, display Multiple VLAN Registration Protocol (MVRP) applicant state information.

Required Privilege Level view

Related Documentation

- show mvrp on page 1397
- show mvrp interface on page 1404
- show mvrp registration-state on page 1406
- show mvrp statistics on page 1408
- show mvrp interface on page 1404
- show mvrp registration-state on page 1406

List of Sample Output show mvrp applicant-state (EX Series and MX Series) on page 1401
show mvrp applicant-state on page 1401

Output Fields Table 156 on page 1400 lists the output fields for the **show mvrp applicant-state** command. Output fields are listed in the approximate order in which they appear.

Table 156: show mvrp applicant-state Output Fields

Field Name	Field Description
VLAN Id	Displays the VLAN ID number.
Interface	Displays the interface number associated with the VLAN ID.

Table 156: show mvrp applicant-state Output Fields (continued)

Field Name	Field Description
State	Displays one of the following MVRP registrar states: • VO— Very anxious observer. • VP —Very anxious passive. • VA —Very anxious new. • AN —Anxious new. • AA —Anxious active. • QA —Quiet active. • LA —Leaving active. • AO —Anxious observer. • QO —Quiet observer. • LO —Leaving observer. • AP —Anxious passive. • QA —Quiet passive.

Sample Output (EX Series and MX Series)

show mvrp applicant-state (EX Series and MX Series)

```
user@host> show mvrp applicant-state
MVRP applicant state for routing instance 'default-switch'
 (VO) Very anxious observer, (VP) Very anxious passive, (VA) Very anxious new,
 (AN) Anxious new, (AA) Anxious active, (QA) Quiet active, (LA) Leaving active,
 (AO) Anxious observer, (QO) Quiet observer, (LO) Leaving observer,
 (AP) Anxious passive, (QP) Quiet passive

VLAN Id     Interface          State
    100     ge-11/3/0          Declaring (QA)
    200     ge-11/3/0          Declaring (QA)
    300     ge-11/3/0          Declaring (QA)
```

Sample Output (SRX1500, SRX300, SRX550M, SRX345, SRX340, and SRX320)

show mvrp applicant-state

```
user@host> show mvrp applicant-state
MVRP applicant state for routing instance 'default-switch'
 (VO) Very anxious observer, (VP) Very anxious passive, (VA) Very anxious new,
 (AN) Anxious new, (AA) Anxious active, (QA) Quiet active, (LA) Leaving active,
 (AO) Anxious observer, (QO) Quiet observer, (LO) Leaving observer,
 (AP) Anxious passive, (QP) Quiet passive

VLAN Id     Interface          State
    1       ge-0/0/1          Idle (VO)
    30      ge-0/0/1          Idle (VO)
    40      ge-0/0/1          Idle (VO)
    50      ge-0/0/1          Idle (VO)
    100     ge-0/0/1          Idle (VO)
```

show mvrp dynamic-vlan-memberships

Syntax show mvrp dynamic-vlan-memberships

Release Information Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 10.1 for MX Series routers.
Command introduced in Junos OS Release 13.2X50-D10 for EX Series switches.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description Display all VLANs that have been created dynamically using Multiple VLAN Registration Protocol (MVRP) on the router, switch, or SRX Series device.

Required Privilege Level clear

Related Documentation

- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Verifying That MVRP Is Working Correctly on Switches on page 548
- **show mvrp on page 1397**
- **show mvrp applicant-state on page 1400**
- **show mvrp interface on page 1404**
- **show mvrp registration-state on page 1406**
- **show mvrp registration-state on page 1406**
- **show mvrp statistics on page 1408**

List of Sample Output **show mvrp dynamic-vlan-memberships (MX Series and EX Series) on page 1403**
show mvrp dynamic-vlan-memberships (EX Series) on page 1403
show mvrp dynamic-vlan-memberships on page 1403

Output Fields Table 157 on page 1402 lists the output fields for the **show mvrp dynamic-vlan-memberships** command on MX Series routers and EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 157: show mvrp dynamic-vlan-memberships Output Fields

Field Name	Field Description
VLAN Id	The VLAN ID of the dynamically created VLAN.
Interfaces	The interface or interfaces that are bound to the dynamically created VLAN.

Sample Output (MX Series Routers and EX Series Switches)

show mvrp dynamic-vlan-memberships (MX Series and EX Series)

```
user@host> show mvrp dynamic-vlan-memberships
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN Id            Interfaces
    100 (s)        ge-11/3/0
    200 (s)        ge-11/3/0
    300 (s)
```

Sample Output (EX Series Switches)

show mvrp dynamic-vlan-memberships (EX Series)

```
user@switch> show mvrp dynamic-vlan-memberships
VLAN Name             Interfaces
-------------------       ----------------
__mvrp_100__              xe-0/1/1.0
                          xe-0/1/0.0
__mvrp_200__              xe-0/1/1.0
                          xe-0/1/0.0
__mvrp_300__              xe-0/1/1.0
```

Sample Output (SRX1500, SRX300, SRX550M, SRX345, SRX340, SRX320)

show mvrp dynamic-vlan-memberships

```
user@host> show mvrp dynamic-vlan-memberships
MVRP dynamic vlans for routing instance 'default-switch'
 (s) static vlan, (f) fixed registration

VLAN Id            Interfaces
    1  (s)
    30 (s)
    40 (s)        ge-0/0/1
    50 (s)        ge-0/0/1
   100 (s)        ge-0/0/1 (f)
```

show mvrp interface

Syntax show mvrp interface

Release Information Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description Display Multiple VLAN Registration Protocol (MVRP) interface-specific information.

Required Privilege Level view

Related Documentation

- show mvrp on page 1397
- show mvrp applicant-state on page 1400
- show mvrp dynamic-vlan-memberships on page 1402
- show mvrp registration-state on page 1406
- show mvrp registration-state on page 1406
- show mvrp statistics on page 1408

List of Sample Output

show mvrp interface on page 1404
show mvrp interface on page 1405

Output Fields Table 158 on page 1404 lists the output fields for the **show mvrp interface** command. Output fields are listed in the approximate order in which they appear.

Table 158: show mvrp interface Output Fields

Field Name	Field Description
Interface	Interface on which MVRP is configured.
Status	Status of the MVRP: **Enabled** or **Disabled**.
Registration Mode	Registration for the interface: **Fixed**, **Forbidden**, or **Normal**.
Applicant Mode	Applicant mode.

Sample Output (MX Series Routers and SX Series Switches)

show mvrp interface

```
user@host> show mvrp interface
MVRP interface information for routing instance 'default-switch'

Interface       Status       Registration      Applicant
                                Mode              Mode
```

```
ge-11/2/8       Enabled      Normal           Normal
ge-11/0/9       Enabled      Normal           Normal
ge-11/3/0       Enabled      Normal           Normal
```

Sample Output (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

show mvrp interface

```
user@host> show mvrp interface
MVRP interface information for routing instance 'default-switch'

Interface       Status       Registration      Applicant
                Mode         Mode              Mode
ge-0/0/1        Enabled      Normal            Normal
```

show mvrp registration-state

Syntax show mvrp registration-state

Release Information Command introduced in Junos OS Release 10.1.
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description For MX Series routers, EX Series switches and SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320, display Multiple VLAN Registration Protocol (MVRP) registration state information.

Required Privilege Level view

Related Documentation

- show mvrp on page 1397
- show mvrp dynamic-vlan-memberships on page 1402
- show mvrp interface on page 1404
- show mvrp statistics on page 1408

List of Sample Output

show mvrp registration-state (EX Series and MX Series) on page 1407
show mvrp registration-state (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320) on page 1407

Output Fields Table 159 on page 1406 lists the output fields for the **show mvrp registration-state** command. Output fields are listed in the approximate order in which they appear.

Table 159: show mvrp registration-state Output Fields

Field Name	Field Description
VLAN Id	Displays the VLAN ID number.
Interface	Displays the interface number associated with the VLAN ID.
Registrar State	Displays whether the registrar state is Registered or Empty.
Forced State	Displays whether the forced state is Registered or Empty.
Managed State	Displays one of the following states: • **fixed**—VLANs always stay in a registered state and are declared as such on all other forwarding ports. • **normal** —VLANs participate in the MVRP protocol and honor incoming join requests normally. • **forbidden** —VLANs ignore the incoming join requests and always stay in an unregistered state.
STP State	Displays whether the Spanning Tree Protocol (STP) is Blocking or Forwarding.

Sample Output

show mvrp registration-state (EX Series and MX Series)

```
user@host> show mvrp registration-state
MVRP registration state for routing instance 'default-switch'

VLAN Id   Interface   Registrar   Forced      Managed    STP
                      State       State       State      State
    100   ge-11/2/8   Empty       Registered  Fixed      Forwarding
          ge-11/0/9   Empty       Empty       Normal     Forwarding
          ge-11/3/0   Registered  Registered  Normal     Forwarding
    101   ge-11/2/8   Empty       Registered  Fixed      Forwarding
          ge-11/0/9   Empty       Empty       Normal     Forwarding
          ge-11/3/0   Registered  Registered  Normal     Forwarding
```

Sample Output

show mvrp registration-state (SRX1500, SRX300, SRX550M, SRX345, SRX340 and SRX320)

```
user@host> show mvrp registration-state
MVRP registration state for routing instance 'default-switch'

VLAN Id   Interface   Registrar   Forced      Managed    STP
                      State       State       State      State
    1     ge-0/0/1    Empty       Empty       Normal     Forwarding
    30    ge-0/0/1    Empty       Empty       Normal     Forwarding
    40    ge-0/0/1    Registered  Registered  Normal     Forwarding
    50    ge-0/0/1    Registered  Registered  Normal     Forwarding
   100    ge-0/0/1    Empty       Registered  Fixed      Forwarding
```

show mvrp statistics

List of Syntax

Syntax (EX Series Switches) on page 1408
Syntax (Switches with ELS Support) on page 1408
Syntax (SRX Devices) on page 1408

Syntax (EX Series Switches)

show mvrp statistics
<interface *interface-name*>

Syntax (Switches with ELS Support)

show mvrp statistics
<interface *interface-name*>
<routing-instance *routing-instance-name*>

Syntax (SRX Devices)

show mvrp statistics

Release Information

Command introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 13.2X50-D10 (ELS).
Command introduced in Junos OS Release 15.1X49-D70 for SRX Series devices.

Description

Display Multiple VLAN Registration Protocol (MVRP) statistics in the form of Multiple Registration Protocol data unit (MRPDU) messages.

Options

none—Show MVRP statistics for all interfaces on the switch.

interface *interface-name*—(Optional) Show MVRP statistics for the specified interface.

Required Privilege Level

view

Related Documentation

- show mvrp on page 1397
- clear mvrp statistics on page 1176
- Example: Configuring Automatic VLAN Administration Using MVRP on EX Series Switches on page 535
- Verifying That MVRP Is Working Correctly on Switches on page 548
- Verifying That MVRP Is Working Correctly on EX Series Switches with ELS Support on page 550

List of Sample Output

show mvrp statistics interface xe-0/1/1.0 on page 1411
show mvrp statistics on page 1411
show mvrp statistics (SRX Devices) on page 1411

Output Fields

Table 157 on page 1402 lists the output fields for the **show mvrp statistics** command on EX Series switches. Output fields are listed in the approximate order in which they appear.

Table 160: show mvrp statistics Output Fields

Field Name	Field Description
MRPDU received	Number of MRPDU messages received on the switch.
Invalid PDU received	Number of invalid MRPDU messages received on the switch.
New received	Number of new messages received on the switch.
Join Empty received	Number of MRP JoinEmpty messages received on the switch. Either this value or the value for *JoinIn received* should increase when the value for *MRPDU received* increases. If this value is not incrementing when it should, you might have a Junos OS release version compatibility issue. To fix a version compatibility issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.
Join In received	Number of MRP JoinIn messages received on the switch. Either this value or the value for *JoinEmpty received* should increase when the value for *MRPDU received* increases. If this value is not incrementing when it should, you might have a Junos OS release version compatibility issue. To fix a version compatibility issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.
Empty received	Number of MRP Empty messages received on the switch.
In received	Number of MRP In messages received on the switch.
Leave received	Number of MRP Leave messages received on the switch.
LeaveAll received	Number of LeaveAll messages received on the switch.
MRPDU transmitted	Number of MRPDU messages transmitted from the switch.
MRPDU transmit failures	Number of MRPDU transmit failures from the switch.
New transmitted	Number of new messages transmitted from the switch.
Join Empty transmitted	Number of JoinEmpty messages sent from the switch.
Join In transmitted	Number of MRP JoinIn messages sent from the switch.
Empty transmitted	Number of MRP Empty messages sent from the switch.
In transmitted	Number of MRP In messages sent from the switch.
Leave transmitted	Number of MRP Leave Empty messages sent from the switch.
LeaveAll transmitted	Number of MRP LeaveAll messages sent from the switch.

Table 161 on page 1410 lists the output fields for the **show mvrp statistics** command on SRX devices. Output fields are listed in the approximate order in which they appear.

Table 161: show mvrp statistics Output Fields

Field Name	Field Description
Interface name	Interface for which MVRP statistics are displayed.
VLAN IDs registered	Number of Virtual LAN (VLAN) IDs registered.
Sent MVRP PDUs	Number of MRPDU messages transmitted from the switch.
Received MVRP PDUs without error	Number of MRPDU messages received on the switch.
Received MVRP PDUs with error	Number of invalid MRPDU messages received on the switch.
Transmitted Join Emply	Number of JoinEmpty messages sent from the switch.
Transmitted Leave All	Number of MRP LeaveAll messages sent from the switch.
Received Join In	Number of MRP JoinIn messages received on the switch. Either this value or the value for **Received Join Empty** should increase when the value for **Received MVRP PDUs without error** increases. If this value is not incrementing when it should, you might have a Junos OS release compatibility issue. To resolve the issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.
Transmitted Join In	Number of MRP JoinIn messages sent from the switch.
Transmitted Empty	Number of MRP Empty messages sent from the switch.
Transmitted Leave	Number of MRP LeaveEmpty messages sent from the switch.
Transmitted In	Number of MRP In messages sent from the switch.
Transmitted New	Number of New messages transmitted from the switch.
Received Leave All	Number of LeaveAll messages received on the switch.
Received Leave	Number of MRP Leave messages received on the switch.
Received In	Number of MRP In messages received on the switch.
Received Empty	Number of MRP Empty messages received on the switch.
Received Join Empty	Number of MRP JoinEmpty messages received on the switch. Either this value or the value for **Received Join In** should increase when the value for **Received MVRP PDUs without error** increases. If this value is not incrementing when it should, you might have a Junos OS release compatibility issue. To resolve the issue, see "Configuring Multiple VLAN Registration Protocol (MVRP) on Switches" on page 504.
Received New	Number of New messages received on the switch.

Sample Output

show mvrp statistics interface xe-0/1/1.0

```
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
```

show mvrp statistics

```
user@host> show mvrp statistics
MVRP statistics for routing instance 'default-switch'

Interface name                  : xe-0/1/1
VLAN IDs registered             : 117
Sent MVRP PDUs                  : 118824
Received MVRP PDUs without error: 118848
Received MVRP PDUs with error   : 0
Transmitted Join Empty          : 5229
Transmitted Leave All           : 2
Recieved Join In                : 11884924
Transmitted Join In             : 1835
Transmitted Empty               : 93606408
Transmitted Leave               : 888
Transmitted In                  : 13780024
Transmitted New                 : 2692
Received Leave All              : 118761
Received Leave                  : 97
Received In                     : 3869
Received Empty                  : 828
Received Join Empty             : 2020152
Received New                    : 224
...
```

show mvrp statistics (SRX Devices)

```
user@host> show mvrp statistics
MVRP statistics for routing instance 'default-switch'

Interface name                  : ge-0/0/1
VLAN IDs registered             : 2
```

```
Sent MVRP PDUs                  : 41
Received MVRP PDUs without error: 28
Received MVRP PDUs with error   : 0
Transmitted Join Empty          : 0
Transmitted Leave All           : 20
Received Join In                : 0
Transmitted Join In             : 0
Transmitted Empty               : 114
Transmitted Leave               : 0
Transmitted In                  : 10
Transmitted New                 : 0
Received Leave All              : 1
Received Leave                  : 0
Received In                     : 0
Received Empty                  : 67
Received Join Empty             : 24
Received New                    : 0
```

show oam ethernet connectivity-fault-management adjacencies

Syntax show oam ethernet connectivity-fault-management adjacencies <interface-name>

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Display connectivity fault management (CFM) adjacencies such as maintenance association end point (MEP) identifier, interface, state of connectivity check protocol, and expiration time.

Options **interface-name**—Dispaly the name of the interface.

Required Privilege Level view

Related Documentation
- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output show oam ethernet connectivity-fault- management adjacencies on page 1413

Output Fields Table 162 on page 1413 lists the output fields for the **show oam ethernet connectivity-fault-management adjacencies** command. Output fields are listed in the approximate order in which they appear

Table 162: show oam ethernet connectivity-fault-management adjacencies Output Fields

Field Name	Field Description
Mep-id	MEP identifier.
Interface	Interface identifier.
State	Indicates whether the connectivity check protocol is up.
Timer to Expire	Indicates the expiration time.

Sample Output

show oam ethernet connectivity-fault- management adjacencies

```
user@host> show oam ethernet connectivity-fault-management adjacencies
Mep-id     Interface        State        Timer to Expire
         101     ge-0/0/4.0         ok                 29
```

show oam ethernet connectivity-fault-management forwarding-state

Syntax show oam ethernet connectivity-fault-management forwarding-state <interface>

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Display the Ethernet Operation, Administration, and Management (OAM) forwarding state for received packets such as interface name, maintenance domain level, maintenance association end point (MEP) direction configured, and next-hop status and index number.

Options **<interface>**—Display the Ethernet OAM state for a forwarding instance.

Required Privilege Level view

Related Documentation
- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output show oam ethernet connectivity-fault- management forwarding-state on page 1415

Output Fields Table 163 on page 1414 lists the output fields for the **show oam ethernet connectivity-fault-management forwarding-state** command. Output fields are listed in the approximate order in which they appear.

Table 163: show oam ethernet connectivity-fault-management forwarding-state Output Fields

Field Name	Field Description
Interface name	Interface identifier.
Level	Maintenance domain level.
Direction	MEP direction configured.
Filter action	Filter action for messages at the maintenance domain level.
Nexthop type	Next-hop type.
Nexthop index	Next-hop index number.

Sample Output

show oam ethernet connectivity-fault- management forwarding-state

```
user@host> show oam ethernet connectivity-fault-management forwarding-state interface
Interface name: ge-0/0/1.0 vlan:100
Instance name: INSTANCE_0 bd_vlan_100
  Maintenance domain forwarding state:

  Level    Direction      Filter action     Nexthop      Nexthop
                                            type         index
  0                       Drop              Discard
  1                       Drop              Discard
  2                       Drop              Discard
  3                       Drop              Discard
  4                       Drop              Discard
  5                       Drop              Discard
  6                       Drop              Discard
  7        down           Receive           Receive
```

show oam ethernet connectivity-fault-management interfaces

Syntax show oam ethernet connectivity-fault-management interfaces <interface name>

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Display Ethernet Operation, Administration, and Management (OAM) information for the specified interface such as link status, maintenance domain level configured, maintenance association end point (MEP) identifier, and MEP neighbors count.

Options **<interface name>**—Display connectivity fault management (CFM) information for the specified interface.

Required Privilege Level view

Related Documentation
- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output show oam ethernet connectivity-fault- management interfaces on page 1417

Output Fields Table 164 on page 1416 lists the output fields for the **show oam ethernet connectivity-fault-management interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 164: show oam ethernet connectivity-fault-management interfaces Output Fields

Field Name	Field Description
Interfaces	Interface identifier.
Link	The local link status is up, down, or oam-down.
Status	The status is active or inactive.
Level	Maintenance domain level configured.
MEP Identifier	MEP identifier.
Neighbors	Number of MEP neighbors.

Sample Output

show oam ethernet connectivity-fault- management interfaces

```
user@host> show oam ethernet connectivity-fault-management interfaces
Interfaces      Link      Status      Level      MEP      Neighbours
                                                        Identifier

        ge-0/0/1.0      Up        Active      7          1000          0
```

show oam ethernet connectivity-fault-management mep-database

Syntax show oam ethernet connectivity-fault-management mep-database

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Display Ethernet Operation, Administration, and Management (OAM) maintenance association end point (MEP) database information.

Options **<local-mep>**—Identifier for the local MEP (1 through 8191).

maintenance-association —Name of the maintenance association.

maintenance-domain —Name of the maintenance domain.

remote-mep —Identifier for the remote MEP (1 through 8191).

Required Privilege Level View

Related Documentation

- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output show oam ethernet connectivity-fault- management mep-database on page 1420

Output Fields Table 165 on page 1418 lists the output fields for the **show oam ethernet connectivity-fault-management mep-database** command. Output fields are listed in the approximate order in which they appear.

Table 165: show oam ethernet connectivity-fault-management mep-database Output Fields

Field Name	Field Description
Maintenance domain name	Maintenance domain name.
Format (Maintenance domain)	Maintenance domain name format configured.
Level	Maintenance domain level configured.
Maintenance association name	Maintenance association name.
Format (Maintenance association)	Maintenance association name format configured.
Continuity-check status	Continuity check status.
Interval	Continuity check message (CCM) interval.

Table 165: show oam ethernet connectivity-fault-management mep-database Output Fields (continued)

Field Name	Field Description
MEP identifier	MEP identifier.
Direction	MEP direction configured.
MAC address	MAC address configured for the MEP.
Auto-discovery	Indicates whether automatic discovery is enabled or disabled.
Priority	Priority used for CCMs and Link Trace Messages (LTMs) transmitted by the MEP.
Interface name	Interface identifier.
Interface status	Local interface status.
Link status	Local link status.
Remote MEP not receiving CCM	Indicates that the remote MEP is not receiving CCMs.
Erroneous CCM received	Indicates that erroneous CCMs have been received.
Cross-connect CCM received	Indicates that cross-connect CCMs have been received.
RDI sent by some MEP	Indicates that the remote defect indication (RDI) bit is set in messages that have been received. The absence of the RDI bit in a CCM indicates that the transmitting MEP is receiving CCMs from all configured MEPs.
CCMs sent	Number of CCMs transmitted.
CCMs received out of sequence	Number of CCMs received out of sequence.
LBMs sent	Number of loopback messages (LBMs) sent.
Valid in-order LBRs received	Number of loopback response (LBR) messages received that were valid messages and in sequence.
Valid out-of-order LBRs received	Number of LBRs received that were valid messages and not in sequence.
LBRs received with corrupted data	Number of LBRs received that were corrupted.
LBRs sent	Number of LBRs transmitted.
LTMs sent	Number of Link Trace Messages (LTMs) transmitted.
LTMs received	Number of LTMs received.
LTRs sent	Number of Link Trace Replies (LTRs) transmitted.
LTRs received	Number of LTRs received.

Table 165: show oam ethernet connectivity-fault-management mep-database Output Fields (continued)

Field Name	Field Description
Sequence number of next LTM request	Sequence number of the next LTM request to be transmitted.
1DMs sent	If the MEP is an initiator for a one-way ETH-DM session, then this is the number of one-way delay measurement (1DM) PDU frames sent to the peer MEP in this session. For all other cases, this field displays 0.
Valid 1DMs received	If the MEP is a receiver for a one-way ETH-DM session, then this is the number of valid 1DM frames received. For all other cases, this field displays 0.
Invalid 1DMs received	If the MEP is a receiver for a one-way ETH-DM session, then this is the number of invalid 1DM frames received. For all other cases, this field displays 0.
DMMs sent	If the MEP is an initiator for a two-way ETH-DM session, then this is the number of delay measurement message (DMM) PDU frames sent to the peer MEP in this session. For all other cases, this field displays 0.
DMRs sent	If the MEP is a responder for a ETH-DM session, then this is the number of delay measurement reply (DMR) frames sent. For all other cases, this field displays 0.
Valid DMRs received	If the MEP is an initiator for a two-way ETH-DM session, then this is the number of valid DMRs received. For all other cases, this field displays 0.
Invalid DMRs received	If the MEP is an initiator for a two-way ETH-DM session, then this is the number of invalid DMRs received. For all other cases, this field displays 0.

Sample Output

show oam ethernet connectivity-fault- management mep-database

```
user@host> show oam ethernet connectivity-fault-management mep-database
maintenance-domain Customer1
Maintenance domain name: Customer1, Format: string, Level: 7
  Maintenance association name: Track_vlan_100, Format: string
  Continuity-check status: enabled, Interval: 1s
  MEP identifier: 1000, Direction: down, MAC address: 2001:db8:5E:00:53:00
  Auto-discovery: disabled, Priority: 0
  Interface name: ge-0/0/1.0, Interface status: Active, Link status: Up
  Defects:
    Remote MEP not receiving CCM                 : no
    Erroneous CCM received                       : no
    Cross-connect CCM received                   : no
```

```
    RDI sent by some MEP                          : no
  Statistics:
    CCMs sent                                     : 170114
    CCMs received out of sequence                 : 0
    LBMs sent                                     : 0
    Valid in-order LBRs received                  : 0
    Valid out-of-order LBRs received              : 0
    LBRs received with corrupted data             : 0
    LBRs sent                                     : 0
    LTMs sent                                     : 0
    LTMs received                                 : 1
    LTRs sent                                     : 1
    LTRs received                                 : 0
    Sequence number of next LTM request           : 0
    1DMs sent                                     : 0
    Valid 1DMs received                           : 0
    Invalid 1DMs received                         : 0
    DMMs sent                                     : 0
    DMRs sent                                     : 0
    Valid DMRs received                           : 0
    Invalid DMRs received                         : 0
Remote MEP count: 1
    Identifier    MAC address        State    Interface
       200     2001:db8:c0:01:01:02       ok    ge-0/0/1.0
    Identifier    MAC address        State    Interface  Timer
```

show oam ethernet connectivity-fault-management mep-statistics

Syntax show oam ethernet connectivity-fault-management mep-statistics
count
local-mep
maintenance-association
maintenance-domain
remote-mep

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Display Ethernet Operation, Administration, and Management (OAM) maintenance end point (MEP) statistics.

NOTE: The delay measurement statistics are not valid for SRX Series devices, which support only the IEEE 802.1ag standard.

Options **count** —Number of statistics per MEP (1 through 100).

local-mep —Identifier for local MEP (1 through 8191).

maintenance-association—Name of maintenance association.

maintenance-domain—Name of maintenance domain.

remote-mep —Identifier for remote MEP (1 through 8191).

Required Privilege Level view

Related Documentation
- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output show oam ethernet connectivity-fault- management mep-statistics on page 1424

Output Fields Table 166 on page 1422 lists the output fields for the **show oam ethernet connectivity-fault-management mep-statistics** command. Output fields are listed in the approximate order in which they appear.

Table 166: show oam ethernet connectivity-fault-management mep-statistics Output Fields

Field Name	Field Description
MEP identifier	Maintenance association end point (MEP) identifier.

Table 166: show oam ethernet connectivity-fault-management mep-statistics Output Fields (continued)

Field Name	Field Description
CCMs sent	Number of CCMs transmitted.
CCMs received out of sequence	Number of CCMs received out of sequence.
LBMs sent	Number of loopback messages (LBMs) sent.
Valid in-order LBRs received	Number of loopback response (LBR) messages received that were valid messages and in sequence.
Valid out-of-order LBRs received	Number of LBRs received that were valid messages and not in sequence.
LBRs received with corrupted data	Number of LBRs received that were corrupted.
LBRs sent	Number of LBRs transmitted.
LTMs sent	Number of Link Trace Messages (LTMs) transmitted.
LTMs received	Number of Link Trace Messages received.
LTRs sent	Number of Link Trace Replies (LTRs) transmitted.
LTRs received	Number of Link Trace responses received.
Sequence number of next LTM request	Sequence number of the next Link Trace Message request to be transmitted.
1DMs sent	If the MEP is an initiator in a one-way ETH-DM session, then this is the number of one-way delay measurement (1DM) PDU frames sent to the peer MEP in this session. For all other cases, this field displays 0.
Valid 1DMs received	If the MEP is a receiver for a one-way ETH-DM session, then this is the number of valid 1DM frames received. For all other cases, this field displays 0.
Invalid 1DMs received	If the MEP is a receiver for a one-way ETH-DM session, then this is the number of invalid 1DM frames received. For all other cases, this field displays 0.
DMMs sent	If the MEP is an initiator for a two-way ETH-DM session, then this is the number of delay measurement message (DMM) PDU frames sent to the peer MEP in this session. For all other cases, this field displays 0.
DMRs sent	If the MEP is a responder for a ETH-DM session, then this is the number of delay measurement reply (DMR) frames sent. For all other cases, this field displays 0.

Table 166: show oam ethernet connectivity-fault-management mep-statistics Output Fields (continued)

Field Name	Field Description
Valid DMRs received	If the MEP is an initiator for a two-way ETH-DM session, then this is the number of valid DMRs received. For all other cases, this field displays 0.
Invalid DMRs received	If the MEP is an initiator for a two-way ETH-DM session, then this is the number of invalid DMRs received. For all other cases, this field displays 0.

Sample Output

show oam ethernet connectivity-fault- management mep-statistics

```
user@host> show oam ethernet connectivity-fault-management mep-statistics
maintenance-domain private maintenance-association private-ma remote-mep 100
MEP identifier: 101, MAC address: 2001:db8:5E:00:53:00
  CCMs sent                                       : 83
  CCMs received out of sequence                   : 0
  LBMs sent                                       : 0
  Valid in-order LBRs received                    : 0
  Valid out-of-order LBRs received                : 0
  LBRs received with corrupted data               : 0
  LBRs sent                                       : 0
  LTMs sent                                       : 0
  LTMs received                                   : 0
  LTRs sent                                       : 0
  LTRs received                                   : 0
  Sequence number of next LTM request             : 0
  1DMs sent                                       : 0
  Valid 1DMs received                             : 0
  Invalid 1DMs received                           : 0
  DMMs sent                                       : 0
  DMRs sent                                       : 0
  Valid DMRs received                             : 0
  Invalid DMRs received                           : 0
```

show oam ethernet connectivity-fault-management mip

Syntax show oam ethernet connectivity-fault-management mip
interface-name
vlan

Release Information Statement introduced in Junos OS Release 12.1X44-D10.

Description Display MIP information.

Options **bridge-domain**—Display information for a particular bridge domain.

instance-name—Display information for a particular routing instance.

interface-name—Display information about the specified logical interface.

vlan—Display information about the specified VLAN (1 through 4094).

Required Privilege Level view

Related Documentation
- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output show oam ethernet connectivity-fault- management mip on page 1425

Output Fields Table 167 on page 1425 lists the output fields for the **show oam ethernet connectivity-fault-management mip** command. Output fields are listed in the approximate order in which they appear.

Table 167: show oam ethernet connectivity-fault-management mip Output Fields

Field Name	Field Description
Default Maintenance-domain	The default maintenance domain name.
Interface	Interface identifier.
Level	Maintenance domain level configured.

Sample Output

show oam ethernet connectivity-fault- management mip

```
user@host> show oam ethernet connectivity-fault-management mip vlan 100
default maintenance-domain mhf    : default

  Interface       Level
```

```
ge-0/0/1.0          5
ge-0/0/4.0          5
```

show oam ethernet connectivity-fault-management path-database

Syntax

show oam ethernet connectivity-fault-management path-database
<host>
maintenance-association
maintenance-domain

Release Information

Statement introduced in Junos OS Release 12.1X44-D10.

Description

Display the Link Trace path database for a remote host.

Options

<host>—MAC address of the remote host in xx:xx:xx:xx:xx:xx format.

maintenance-association —Name of the maintenance association.

maintenance-domain —Name of the maintenance domain.

Required Privilege Level

view

Related Documentation

- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179

List of Sample Output

show oam ethernet connectivity-fault- management path-database on page 1428

Output Fields

Table 168 on page 1427 lists the output fields for the **show oam ethernet connectivity-fault-management path-database** command. Output fields are listed in the approximate order in which they appear.

Table 168: show oam ethernet connectivity-fault-management path-database Output Fields

Field Name	Field Description
Interface	Interface Identifier.
Maintenance Domain	Maintenance domain name.
Maintenance Association	Maintenance association name.
Level	Maintenance domain level configured for the maintenance domain.
Hop	Sequential hop count of the Link Trace path.
TTL	Number of hops remaining in the Link Trace message (LTM). The time to live (TTL) is decremented at each hop.
Source MAC Address	MAC address of the 802.1ag maintenance association intermediate point (MIP) that is forwarding the LTM.

Table 168: show oam ethernet connectivity-fault-management path-database Output Fields (continued)

Field Name	Field Description
Next-hop MAC Address	MAC address of the 802.1ag node that is the next hop in the LTM path.
Transaction Identifier	Identifier maintained by the MEP. Each LTM uses a transaction identifier. The transaction identifier is maintained globally across all maintenance domains. Use the transaction identifier to match an incoming Link Trace Reply (LTR) with a previously sent LTM.

Sample Output

show oam ethernet connectivity-fault- management path-database

```
user@host> show oam ethernet connectivity-fault-management path-database
Interface : ge-0/0/4
      Maintenance Domain: private, Level: 5
      Maintenance Association: private-ma, Local Mep: 100

    Hop   TTL    Source MAC address        Next-hop MAC address
    Transaction Identifier:0
    1     63     00:00:5E:00:53:AA        00:00:5E:00:53:AB
    2     62     00:00:5E:00:53:AC         00:00:5E:00:53:AD
    Transaction Identifier:1
    1     63     00:00:5E:00:53:AE         00:00:5E:00:53:AF
    2     62     00:00:5E:00:53:AG         00:00:5E:00:53:AH
    Transaction Identifier:2
    1     63     00:00:5E:00:53:AI         00:00:5E:00:53:AJ
    2     62     00:00:5E:00:53:AK         00:00:5E:00:53:AL
    Transaction Identifier:3
    1     63     00:00:5E:00:53:AM         00:00:5E:00:53:AN
    2     62     00:00:5E:00:53:AO         00:00:5E:00:53:AP
```

show oam ethernet link-fault-management

Syntax
show oam ethernet link-fault-management
<brief | detail>
<interface-name>

Release Information
Statement for SRX Series devices introduced in Junos OS Release 9.5.

Description
Display Operation, Administration, and Maintenance (OAM) link fault management (LFM) information for Ethernet interfaces.

Options
brief | detail—(Optional) Display the specified level of output.

interface-name—(Optional) Display link fault management information for the specified Ethernet interface only.

Required Privilege Level
view

Related Documentation
- clear oam ethernet connectivity-fault-management path-database on page 1178
- clear oam ethernet connectivity-fault-management statistics on page 1179
- Understanding Ethernet OAM Link Fault Management for SRX Series Services Gateways on page 809
- Example: Configuring Ethernet OAM Link Fault Management on a Security Device on page 811

List of Sample Output
show oam ethernet link-fault-management brief on page 1433
show oam ethernet link-fault-management detail on page 1433

Output Fields
Table 169 on page 1429 lists the output fields for the **show oam ethernet link-fault-management** command. Output fields are listed in the approximate order in which they appear.

Table 169: show oam ethernet link-fault-management Output Fields

Field Name	Field Description	Level of Output
Status	Status of the established link. • **Fail**—A link fault condition exists. • **Running**—A link fault condition does not exist.	All levels

Table 169: show oam ethernet link-fault-management Output Fields (continued)

Field Name	Field Description	Level of Output
Discovery state	State of the discovery mechanism: • **Passive Wait** • **Send Any** • **Send Local Remote** • **Send Local Remote Ok**	All levels
Peer address	Address of the OAM peer.	All levels
Flags	Information about the interface. • **Remote-Stable**—Indicates remote OAM client acknowledgment of, and satisfaction with, local OAM state information. **False** indicates that remote DTE has either not seen or is unsatisfied with local state information. **True** indicates that remote DTE has seen and is satisfied with local state information. • **Local-Stable**—Indicates local OAM client acknowledgment of, and satisfaction with, remote OAM state information. **False** indicates that local DTE either has not seen or is unsatisfied with remote state information. **True** indicates that local DTE has seen and is satisfied with remote state information. • **Remote-State-Valid**—Indicates the OAM client has received remote state information found within local information TLVs (type, length, values) of received Information OAM PDUs. **False** indicates that the OAM client has not seen remote state information. **True** indicates that the OAM client has seen remote state information.	All levels
Remote loopback status	An OAM entity can put its remote peer into loopback mode using the Loopback control OAM PDU. In loopback mode, every frame received is transmitted back on the same port (except for OAM PDUs, which are needed to maintain the OAM session).	All levels
Remote entity information	Remote entity information. • **Remote MUX action**—Indicates the state of the multiplexer functions of the OAM sublayer. Device is forwarding non-OAM PDUs to the lower sublayer or discarding non-OAM PDUs. • **Remote parser action**—Indicates the state of the parser function of the OAM sublayer. Device is forwarding non-OAM PDUs to the higher sublayer, looping back non-OAM PDUs to the lower sublayer, or discarding non-OAM PDUs. • **Discovery mode**—Indicates whether discovery mode is active or inactive. • **Unidirectional mode**—Indicates the ability to operate a link in unidirectional mode for diagnostic purposes. • **Remote loopback mode**—Indicates whether remote loopback is supported or not supported. • **Link events**—Indicates whether interpreting link events is supported or not supported on the remote peer. • **Variable requests**—Indicates whether variable requests are supported or not supported. The Variable Request OAM PDU, is used to request one or more MIB variables from the remote peer.	All levels
OAM Receive Statistics		

Table 169: show oam ethernet link-fault-management Output Fields (continued)

Field Name	Field Description	Level of Output
Information	Number of information PDUs received.	**detail**
Event	Number of loopback control PDUs received.	**detail**
Variable request	Number of variable request PDUs received.	**detail**
Variable response	Number of variable response PDUs received.	**detail**
Loopback control	Number of loopback control PDUs received.	**detail**
Organization specific	Number of vendor organization specific PDUs received.	**detail**
OAM Transmit Statistics		
Information	Number of information PDUs transmitted.	**detail**
Event	Number of event notification PDUs transmitted.	**detail**
Variable request	Number of variable request PDUs transmitted.	**detail**
Variable response	Number of variable response PDUs transmitted.	**detail**
Loopback control	Number of loopback control PDUs transmitted.	**detail**
Organization specific	Number of vendor organization specific PDUs transmitted.	**detail**
OAM Received Symbol Error Event information		
Events	Number of symbol error event TLVs that have been received after the OAM sublayer was reset.	**detail**
Window	Symbol error event window in the received PDU. The protocol default value is the number of symbols that can be received in one second on the underlying physical layer.	**detail**
Threshold	Number of errored symbols in the period required for the event to be generated.	**detail**
Errors in period	Number of symbol errors in the period reported in the received event PDU.	**detail**
Total errors	Number of errored symbols that have been reported in received event TLVs after the OAM sublayer was reset. Symbol errors are coding symbol errors.	**detail**
OAM Received Frame Error Event Information		
Events	Number of errored frame event TLVs that have been received after the OAM sublayer was reset.	**detail**

Table 169: show oam ethernet link-fault-management Output Fields (continued)

Field Name	Field Description	Level of Output
Window	Duration of the window in terms of the number of 100 ms period intervals.	**detail**
Threshold	Number of detected errored frames required for the event to be generated.	**detail**
Errors in period	Number of detected errored frames in the period.	**detail**
Total errors	Number of errored frames that have been reported in received event TLVs after the OAM sublayer was reset. A frame error is any frame error on the underlying physical layer.	**detail**
OAM Received Frame Period Error Event Information		
Events	Number of frame seconds errors event TLVs that have been received after the OAM sublayer was reset.	**detail**
Window	Duration of the frame seconds window.	**detail**
Threshold	Number of frame seconds errors in the period.	**detail**
Errors in period	Number of frame seconds errors in the period.	**detail**
Total errors	Number of frame seconds errors that have been reported in received event TLVs after the OAM sublayer was reset.	**detail**
OAM Transmitted Symbol Error Event Information		
Events	Number of symbol error event TLVs that have been transmitted after the OAM sublayer was reset.	**detail**
Window	The symbol error event window in the transmitted PDU.	**detail**
Threshold	Number of errored symbols in the period required for the event to be generated.	**detail**
Errors in period	Number of symbol errors in the period reported in the transmitted event PDU.	**detail**
Total errors	Number of errored symbols reported in event TLVs that have been transmitted after the OAM sublayer was reset.	**detail**
OAM Transmitted Frame Error Event Information		
Events	Number of errored frame event TLVs that have been transmitted after the OAM sublayer was reset.	**detail**
Window	Duration of the window in terms of the number of 100-ms period intervals.	**detail**
Threshold	Number of detected errored frames required for the event to be generated.	**detail**
Errors in period	Number of detected errored frames in the period.	**detail**

Table 169: show oam ethernet link-fault-management Output Fields (continued)

Field Name	Field Description	Level of Output
Total errors	Number of errored frames that have been detected after the OAM sublayer was reset.	**detail**

Sample Output

show oam ethernet link-fault-management brief

```
user@host> show oam ethernet link-fault-management brief
Interface: ge-0/0/1
  Status: Running, Discovery state: Send Any
  Peer address: 2001:bd8:00:31
  Flags:Remote-Stable Remote-State-Valid Local-Stable 0x50
  Remote loopback status: Disabled on local port, Enabled on peer port
  Remote entity information:
    Remote MUX action: discarding, Remote parser action: loopback
    Discovery mode: active, Unidirectional mode: unsupported
    Remote loopback mode: supported, Link events: supported
    Variable requests: unsupported
```

show oam ethernet link-fault-management detail

```
user@host> show oam ethernet link-fault-management detail
  Interface: ge-0/0/1
    Status: Running, Discovery state: Send Any
    Peer address: 2001:bd8:00:31
    Flags:Remote-Stable Remote-State-Valid Local-Stable 0x50
    OAM receive statistics:
      Information: 186365, Event: 0, Variable request: 0, Variable response: 0
      Loopback control: 0, Organization specific: 0
    OAM transmit statistics:
      Information: 186347, Event: 0, Variable request: 0, Variable response: 0
      Loopback control: 0, Organization specific: 0
    OAM received symbol error event information:
      Events: 0, Window: 0, Threshold: 0
      Errors in period: 0, Total errors: 0
    OAM received frame error event information:
      Events: 0, Window: 0, Threshold: 0
      Errors in period: 0, Total errors: 0
    OAM received frame period error event information:
      Events: 0, Window: 0, Threshold: 0
      Errors in period: 0, Total errors: 0
    OAM transmitted symbol error event information:
      Events: 0, Window: 0, Threshold: 1
      Errors in period: 0, Total errors: 0
    OAM transmitted frame error event information:
      Events: 0, Window: 0, Threshold: 1
      Errors in period: 0, Total errors: 0
    Remote entity information:
      Remote MUX action: forwarding, Remote parser action: forwarding
      Discovery mode: active, Unidirectional mode: unsupported
      Remote loopback mode: supported, Link events: supported
      Variable requests: unsupported
```

show protection-group ethernet-ring aps

Syntax **show protection-group ethernet-ring aps**

Release Information Command introduced in Junos OS Release 9.4.
Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description Display the status of the Automatic Protection Switching (APS) and Ring APS (RAPS) messages on an Ethernet ring.

Options This command has no options.

Required Privilege Level view

Related Documentation

- show protection-group ethernet-ring data-channel on page 1444
- show protection-group ethernet-ring interface on page 1447
- show protection-group ethernet-ring node-state on page 1451
- show protection-group ethernet-ring statistics on page 1456
- show protection-group ethernet-ring vlan on page 1462

List of Sample Output

Output Fields Table 170 on page 1435 lists the output fields for the **show protection-group ethernet-ring aps** command. Output fields are listed in the approximate order in which they appear.

Table 170: show protection-group ethernet-ring aps Output Fields

Field Name	Field Description
Ethernet Ring	Name configured for the Ethernet ring.
Request/State	Status of the Ethernet ring RAPS messages. • **NR**—Indicates that there is no request for APS on the ring. • **SF**—Indicates that there is a signal failure on the ring. • **FS**—Indicates that there are active forced-switch requests in the ring. • **MS**—Indicates that there are active manual-switch requests in the ring. NOTE: Both **FS** and **MS** values are valid only when G.8032v2 is supported.
Ring Protection Link Blocked	Blocking on the ring protection link: **Yes** or **No**.
No Flush	Indicates the value of the Do Not Flush (DNF) flag in the received RAPS PDU. If the value is Yes, then FDB flush is not triggered as part of processing of the received RAPS PDU.
Blocked Port Reference	This parameter is the reference to the blocked ring port. If the east ring port is blocked, the Blocked Port Reference (BPR) value is 0. If the west ring port is blocked, the BPR value is 1. If both ring ports are blocked, this parameter can take any value. If both east and west ports are blocked or not blocked, the value would be 0. This field is valid only when G.8032v2 is supported.
Blocked Port Reference	Reference of the ring port on which traffic is blocked.
Originator	Indicates whether the node is the originator of the RAPS messages.
Remote Node ID	Identifier (in MAC address format) of the remote node.

Sample Output

show protection-group ethernet-ring aps (EX Switches)

```
user@switch>show protection-group ethernet-ring aps
Ring Name    Request/state  No Flush  RPL Blocked  Originator  Remote Node ID
erp1         NR             No        Yes          No          00:1F:12:30:B8:81
```

Sample Output

show protection-group ethernet-ring aps (Owner Node, Normal Operation on ACX and MX Routers)

```
user@host> show protection-group ethernet-ring aps
Ethernet Ring   Request/state  RPL Blocked  No Flush  BPR  Originator  Remote
Node ID
Erp_1           NR             Yes          No        1    No
00:00:00:02:00:01
```

Sample Output

show protection-group ethernet-ring aps detail (Owner Node, Normal Operation on ACX and MX Routers)

```
user@host> show protection-group ethernet-ring aps detail
Ethernet-Ring name              : Erp_1
Request/State                   : NR
Ring Protection Link blocked    : Yes
No Flush Flag                   : No
Blocked Port Reference          : 1
Originator                      : No
Remote Node ID                  : 00:00:00:02:00:01
```

show protection-group ethernet-ring aps (MX RPL Owner Ring Node, Failure condition on non-RPL link of the ring)

```
user@host>show protection-group ethernet-ring aps
Ethernet Ring   Request/state   RPL Blocked  No Flush
pg101           SF              No           No

Originator  Remote Node ID
No          00:01:02:00:00:01
```

show protection-group ethernet-ring aps (MX Interconnection Ring Node, Failure condition in major ring on non-RPL link of the ring)

```
user@host>show protection-group ethernet-ring aps
Ethernet Ring   Request/state  RPL Blocked  No Flush  BPR
pg_major        SF             No           No        0
pg_subring      NR             Yes          Yes       0

Originator  Remote Node ID
No          00:01:00:00:00:01
No          00:02:00:00:00:02
```

show protection-group ethernet-ring aps (MX Series router)

```
user@host>show protection-group ethernet-ring aps
Ethernet Ring   Request/state  RPL Blocked  No Flush  BPR  Originator  Remote
Node ID
Inst_Vlans_1-15 NR             Yes          Yes       1    Yes         NA

Inst_Vlans_16-30 NR            Yes          Yes       0    No
00:00:00:03:00:02
```

show protection-group ethernet-ring aps detail (MX Series router)

```
user@host>show protection-group ethernet-ring aps
Ethernet-Ring name              : Inst_Vlans_1-15
Request/State                   : NR
Ring Protection Link blocked    : Yes
No Flush Flag                   : Yes
Blocked Port Reference          : 1
Originator                      : Yes
Remote Node ID                  : NA

Ethernet-Ring name              : Inst_Vlans_16-30
Request/State                   : NR
Ring Protection Link blocked    : Yes
```

```
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : No
Remote Node ID                 : 00:00:00:03:00:02
```

show protection-group ethernet-ring aps (MX Interconnection Ring Node as RPL owner of major ring, rings in IDLE state)

```
user@host>show protection-group ethernet-ring aps detail

Ethernet-Ring name             : pg_major
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : Yes
Remote Node ID                 : NA

Ethernet-Ring name             : pg_subring
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : No
Remote Node ID                 : 00:00:03:00:00:03
```

show protection-group ethernet-ring aps detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring aps detail
Ethernet-Ring name             : pg1001
Request/State                  : NR
Ring Protection Link blocked   : Yes
No Flush Flag                  : Yes
Blocked Port Reference         : 0
Originator                     : Yes
Remote Node ID                 : NA
```

show protection-group ethernet-ring configuration

Syntax **show protection-group ethernet-ring configuration**

Release Information Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 14.1 for MX Series routers.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description Display the configuration of Ethernet ring protection group on EX Switches and MX Series routers.

Required Privilege Level view

Related Documentation

- show protection-group ethernet-ring aps on page 1434
- show protection-group ethernet-ring data-channel on page 1444
- show protection-group ethernet-ring interface on page 1447
- show protection-group ethernet-ring node-state on page 1451
- show protection-group ethernet-ring statistics on page 1456
- show protection-group ethernet-ring vlan on page 1462

List of Sample Output

show protection-group ethernet-ring configuration (EX Switch) on page 1440
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1441
show protection-group ethernet-ring configuration (MX Series Router) on page 1441
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1441
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1442
show protection-group ethernet-ring configuration (MX Series Router) on page 1442
show protection-group ethernet-ring configuration detail (MX Series Router) on page 1443

Output Fields Table 171 on page 1438 lists the output fields for the **show protection-group ethernet-ring configuration** command. Output fields are listed in the approximate order in which they appear.

Table 171: show protection-group ethernet-ring configuration Output Fields

Output Fields	Field Description
G8032 Compatability Version	This is the compatibility version mode of ERP. This parameter always takes the value 1 in the case of G8032v1. This parameter is valid only for MX Series routers.
East Interface	One of the two switch interfaces that participates in a ring link. When Junos supports G8032v2, this interface is treated as interface 0.
West Interface	One of the two interfaces in a switch that participates in a ring link. When Junos supports G8032v2, this interface is treated as interface 1.

Table 171: show protection-group ethernet-ring configuration Output Fields (continued)

Output Fields	Field Description
Restore Interval	Configured interval of wait time after a link is restored. When a link goes down, the RPL link is activated. When the down link becomes active again, the RPL owner receives a notification. The RPL owner waits for the restore interval before issuing a block on the RPL link. The configured restore interval can be 5 through 12 minutes for ERPv1 and 1 through 12 minutes for ERPv2. This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value. NOTE: Wait to Restore (WTR) configuration values on EX2300 and EX3400 switches must be 5-12 minutes.
Wait to Block Interval	Configured interval of wait time for link restoration when a manual command (manual switch or force switch) is cleared. On clearing the manual command, the RPL owner receives NR messages, which starts a timer with interval 'Wait to Block' to restore the RPL link after its expiration. This delay timer is set to be 5 seconds longer than the guard timer. The configured number can be from 5 seconds through 10 seconds. The parameter is valid only for G.8032v2. NOTE: The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.
Guard Interval	Configured number of milliseconds (in 10 millisecond intervals, 10 milliseconds through 2000 milliseconds) that the node does not process any Ethernet ring protection protocol data units (PDUs). This configuration is a global configuration and applies to all Ethernet rings if the Ethernet ring does not have a more specific configuration for this value. If no parameter is configured at the protection group level, the global configuration of this parameter uses the default value.
Hold off interval	This is the interval at which the link is held down even before declaring that the link is down. Because the parameter is not supported at present, its value is always considered 0. This parameter is valid only for MX Series routers.
Node ID	Node ID for the switch or router. If the node ID is not configured, it is assigned by default. For EX Series switches, the Node ID value cannot be configured, whereas for MX Series routers, it can be configured.
Ring ID	In G8032v2, the ring ID can be within the range 1–239. All the nodes in a ring should have the same ring ID. In the case of G8032v1, the value of the ring ID is always 1. This parameter is valid only for MX Series routers.
Node Role	Indicates whether the ring node is operating as a normal ring-node or RPL-owner or RPL-neighbor. For G8032v1 RPL-neighbor role is not supported. This parameter is valid only for MX Series routers.

Table 171: show protection-group ethernet-ring configuration Output Fields (continued)

Output Fields	Field Description
Revertive Mode of Operation	This parameter indicates whether the ring is operating in revertive mode or nonrevertive mode. In nonrevertive mode of operation, when all links in the ring and Ethernet Ring Nodes have recovered and no external requests are active, the Ethernet Ring does not automatically revert. G8032v1 supports only revertive mode of operation. This parameter is valid only for MX Series routers.
RAPS Tx Dot1p priority	The RAPS Tx Dot1p priority is a parameter with which the RAPS is transmitted from the ring node. For G8032v1, the value of this parameter is always 0. For G8032v2, the value of this parameter can be within the range 0–7. This parameter is valid only for MX Series routers.
Node type	Indicates whether ring node is a normal ring node having two ring-links or a open ring-node having only a single ring-link or a interconnection ring-node. An interconnection ring node can be connected to major ring in non virtual-channel mode or in virtual channel mode. Ring interconnection is not supported for G8032v1. This parameter is valid only for MX Series routers.
Major ring name	If the node type is interconnection in the ring, this parameter takes the name of the major ring to which the sub-ring node is connected. This parameter is valid only for MX Series routers.
Interconnection mode	Indicates the interconnection mode if the type of the node is interconnection. An interconnection ring node can be connected to major ring in non-virtual channel mode or in virtual channel mode. This parameter is valid only for MX Series routers.
Propagate Topology Change event	When Propagate Topology Change event is set to 1, the change in the topology of sub-ring is propagated to the major ring, enabling the transmission of EVENT FLUSH RAPS PDU in the major ring. When the parameter is set to 0, the topology change in the sub-ring is not propagated to the major ring blocking EVENT FLUSH RAPS PDU transmission in the major ring. This parameter is valid only for MX Series routers.
Control Vlan	The VLAN that transfers ERP PDUs from one node to another.
Physical Ring	Physical ring if the east and west interfaces are nontrunk ports. For MX Series routers, the ring is termed a physical ring if no data channels are defined for the ring and the entire physical port forwarding is controlled by ERP.
Data Channel VLAN(s)	Data VLANs for which forwarding behavior is controlled by the ring instance.

Sample Output

show protection-group ethernet-ring configuration (EX Switch)

```
user@switch>show protection-group ethernet-ring configuration
Ethernet ring configuration parameters for protection group erp1
East Interface   : ge-0/0/3.0
West Interface   : ge-0/0/9.0
Restore Interval : 5 minutes
Guard Interval   : 500 ms
Node Id          : 00:1F:12:30:B8:81
```

```
Control Vlan     : 101
Physical Ring    : yes
```

show protection-group ethernet-ring configuration detail (MX Series Router)

```
user@switch>show protection-group ethernet-ring configuration detail
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                 : 2
East interface (interface 0)                : xe-2/3/0.1
West interface (interface 1)                : xe-2/2/1.1
Restore interval                            : 5 minutes
Wait to Block interval                      : 5 seconds
Guard interval                              : 500 ms
Hold off interval                           : 0 ms
Node ID                                     : 64:87:88:65:37:D0
Ring ID (1 ... 239)                         : 1
Node role (normal/rpl-owner/rpl-neighbour)  : normal
Revertive mode of operation                 : 1
RAPS Tx Dot1p priority (0 .. 7)             : 0
Node type (normal/open/interconnection)     : Normal
Control Vlan                                : 100
Physical Ring                               : No
Data Channel Vlan(s)                        : 200,300
```

show protection-group ethernet-ring configuration (MX Series Router)

```
user@switch>show protection-group ethernet-ring configuration
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                 : 2
East interface (interface 0)                : xe-2/3/0.1
West interface (interface 1)                : xe-2/2/1.1
Restore interval                            : 5 minutes
Wait to Block interval                      : 5 seconds
Guard interval                              : 500 ms
Hold off interval                           : 0 ms
Node ID                                     : 64:87:88:65:37:D0
Ring ID (1 ... 239)                         : 1
Node role (normal/rpl-owner/rpl-neighbour)  : rpl-neighbour
Node RPL end                                : east-port
Revertive mode of operation                 : 1
RAPS Tx Dot1p priority (0 .. 7)             : 0
Node type (normal/open/interconnection)     : Normal
Control Vlan                                : 100
Physical Ring                               : No
Data Channel Vlan(s)                        : 200,300
```

show protection-group ethernet-ring configuration detail (MX Series Router)

```
user@switch>show protection-group ethernet-ring configuration detail
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                 : 2
East interface (interface 0)                : xe-2/3/0.1
West interface (interface 1)                : xe-2/2/1.1
Restore interval                            : 5 minutes
Wait to Block interval                      : 5 seconds
Guard interval                              : 500 ms
Hold off interval                           : 0 ms
Node ID                                     : 64:87:88:65:37:D0
Ring ID (1 ... 239)                         : 1
Node role (normal/rpl-owner/rpl-neighbour)  : rpl-owner
```

```
Node RPL end                                  : east-port
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Normal
Control Vlan                                  : 100
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300
```

show protection-group ethernet-ring configuration detail (MX Series Router)

```
user@switch>show protection-group ethernet-ring configuration detail
Ethernet Ring configuration information for protection group pg_101
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : xe-2/3/0.1
West interface (interface 1)                  : (no erp)
Restore interval                              : 5 minutes
Wait to Block interval                        : 5 seconds
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
Ring ID (1 ... 239)                           : 1
Node role (normal/rpl-owner/rpl-neighbour)    : rpl-owner
Node RPL end                                  : east-port
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Open
Control Vlan                                  : 100
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300
```

show protection-group ethernet-ring configuration (MX Series Router)

```
user@switch>show protection-group ethernet-ring configuration
Ethernet Ring configuration information for protection group pg_major
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : xe-2/3/0.1
West interface (interface 1)                  : xe-2/2/1.1
Restore interval                              : 5 minutes
Wait to Block interval                        : 5 seconds
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
Ring ID (1 ... 239)                           : 1
Node role (normal/rpl-owner/rpl-neighbour)    : rpl-owner
Node RPL end                                  : east-port
Revertive mode of operation                   : 1
RAPS Tx Dot1p priority (0 .. 7)               : 0
Node type (normal/open/interconnection)       : Normal
Control Vlan                                  : 100
Physical Ring                                 : No
Data Channel Vlan(s)                          : 200,300

Ethernet Ring configuration information for protection group pg_subring
G8032 Compatibility Version                   : 2
East interface (interface 0)                  : ge-2/0/0.1
West interface (interface 1)                  : (no erp)
Restore interval                              : 5 minutes
Wait to Block interval                        : 5 seconds
Guard interval                                : 500 ms
Hold off interval                             : 0 ms
Node ID                                       : 64:87:88:65:37:D0
```

```
Ring ID (1 ... 239)                          : 2
Node role (normal/rpl-owner/rpl-neighbour)   : normal
Revertive mode of operation                  : 1
RAPS Tx Dot1p priority (0 .. 7)              : 0
Node type (normal/open/interconnection)      : Non-VC-Interconnection
Major ring name                              : pg_major
Interconnection mode (VC/Non-VC)             : Non-VC mode
Propagate Topology Change event              : 0
Control Vlan                                 : 101
Physical Ring                                : No
Data Channel Vlan(s)                         : 200,300
```

show protection-group ethernet-ring configuration detail (MX Series Router)

```
user@switch>show protection-group ethernet-ring configuration detail
Ethernet Ring configuration information for protection group pg_major
G8032 Compatibility Version                  : 2
East interface (interface 0)                 : xe-2/3/0.1
West interface (interface 1)                 : xe-2/2/1.1
Restore interval                             : 5 minutes
Wait to Block interval                       : 5 seconds
Guard interval                               : 500 ms
Hold off interval                            : 0 ms
Node ID                                      : 64:87:88:65:37:D0
Ring ID (1 ... 239)                          : 1
Node role (normal/rpl-owner/rpl-neighbour)   : rpl-owner
Node RPL end                                 : east-port
Revertive mode of operation                  : 1
RAPS Tx Dot1p priority (0 .. 7)              : 0
Node type (normal/open/interconnection)      : Normal
Control Vlan                                 : 100
Physical Ring                                : No
Data Channel Vlan(s)                         : 200,300

Ethernet Ring configuration information for protection group pg_subring
G8032 Compatibility Version                  : 2
East interface (interface 0)                 : ge-2/0/0.1
West interface (interface 1)                 : (no erp)
Restore interval                             : 5 minutes
Wait to Block interval                       : 5 seconds
Guard interval                               : 500 ms
Hold off interval                            : 0 ms
Node ID                                      : 64:87:88:65:37:D0
Ring ID (1 ... 239)                          : 2
Node role (normal/rpl-owner/rpl-neighbour)   : normal
Revertive mode of operation                  : 1
RAPS Tx Dot1p priority (0 .. 7)              : 0
Node type (normal/open/interconnection)      : Non-VC-Interconnection
Major ring name                              : pg_major
Interconnection mode (VC/Non-VC)             : Non-VC mode
Propagate Topology Change event              : 0
Control Vlan                                 : 101
Physical Ring                                : No
Data Channel Vlan(s)                         : 200,300
```

show protection-group ethernet-ring data-channel

Syntax

show protection-group ethernet-ring data-channel
<brief | detail>
<group-name *group-name*>

Release Information

Command introduced in Junos OS Release 10.2.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

Display the configuration of Ethernet ring protection group on EX Switches and MX Series routers.

Options

brief | detail—(Optional) Display the specified level of output.

group-name—(Optional) Protection group for which to display statistics. If you omit this optional field, all protection group statistics for configured groups will be displayed.

Required Privilege Level

view

Related Documentation

- show protection-group ethernet-ring aps on page 1434
- show protection-group ethernet-ring interface on page 1447
- show protection-group ethernet-ring node-state on page 1451
- show protection-group ethernet-ring statistics on page 1456
- show protection-group ethernet-ring vlan on page 1462

List of Sample Output

show protection-group ethernet-ring data-channel on page 1445
show protection-group ethernet-ring data-channel detail on page 1445
show protection-group ethernet-ring data-channel detail (EX2300 and EX3400 Switches) on page 1446

Output Fields

Table 172 on page 1444 lists the output fields for the **show protection-group ethernet-ring data-channel** command. Output fields are listed in the approximate order in which they appear.

Table 172: show protection-group ethernet-ring data-channel Output Fields

Field Name	Field Description
Interface	Name of the interface configured for the Ethernet ring.

Table 172: show protection-group ethernet-ring data-channel Output Fields (continued)

Field Name	Field Description
STP index	The Spanning Tree Protocol (STP) index number used by each interface in an Ethernet ring. The STP index controls the forwarding behavior for a set of VLANs on a data channel on an Ethernet ring port. For multiple Ethernet ring instances on an physical ring port, there are multiple STP index numbers. Different ring instances will have different STP index numbers and may have different forwarding behavior.
Forward State	Forwarding state on the Ethernet ring. • **fowarding**—Indicates packets are being forwarded. • **discarding**—Indicates packets are being discarded.

Sample Output

show protection-group ethernet-ring data-channel

```
user@host> show protection-group ethernet-ring data-channel
Ethernet ring data channel information for protection group pg301

Interface    STP index  Forward State
xe-5/0/2     78         forwarding
xe-2/2/0     79         discarding

Ethernet ring data channel parameters for protection group pg302

Interface    STP index  Forward State
xe-5/0/2     80         forwarding
xe-2/2/0     81         forwarding
```

show protection-group ethernet-ring data-channel detail

```
user@host> show protection-group ethernet-ring data-channel detail
Ethernet ring data channel parameters for protection group pg301

Interface name                : xe-5/0/2
STP index                     : 78
Forward State                 : forwarding

Interface name                : xe-2/2/0
STP index                     : 79
Forward State                 : discarding


Ethernet ring data channel parameters for protection group pg302

Interface name                : xe-5/0/2
STP index                     : 80
Forward State                 : forwarding

Interface name                : xe-2/2/0
STP index                     : 81
Forward State                 : forwarding
```

show protection-group ethernet-ring data-channel detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring data-channel detail
Ethernet ring data channel parameters for protection group pg1001

Interface name                : ge-0/0/42
STP index                     : 52
Forward State                 : discarding

Interface name                : ge-0/0/38
STP index                     : 53
Forward State                 : forwarding
```

show protection-group ethernet-ring interface

Syntax **show protection-group ethernet-ring interface**

Release Information Command introduced in Junos OS Release 9.4.
Command introduced in Junos OS Release 12.3X54 for ACX Series routers.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description Displays the status of the Automatic Protection Switching (APS) interfaces on an Ethernet ring.

Options This command has no options.

Required Privilege Level view

Related Documentation

- show protection-group ethernet-ring data-channel on page 1444
- show protection-group ethernet-ring aps on page 1434
- show protection-group ethernet-ring node-state on page 1451
- show protection-group ethernet-ring statistics on page 1456
- show protection-group ethernet-ring vlan on page 1462

List of Sample Output

Output Fields Table 173 on page 1448 lists the output fields for both the EX Series switch, and the ACX Series and MX Series router **show protection-group ethernet-ring interface** commands. Output fields are listed in the approximate order in which they appear.

Table 173: MX Series Routers show protection-group ethernet-ring interface Output Fields

Field Name	Field Description
Ethernet ring port parameters for protection group *group-name*	Output is organized by configured protection group.
Interface	Physical interfaces configured for the Ethernet ring. This can be an aggregated Ethernet link also.
Control Channel	(MX Series router only) Logical unit configured on the physical interface.
Direction	Direction of the traffic.
Forward State	State of the ring forwarding on the interface: **discarding** or **forwarding**.
Ring Protection Link End	Whether this interface is the end of the ring: **Yes** or **No**.
Signal Failure	Whether there a signal failure exists on the link: **Clear** or **Set**.
Admin State	State of the interface: For EX switches, **ready**, **ifl ready**, or **waiting**. For MX routers, **IFF ready** or **IFF disabled**.

Sample Output

show protection-group ethernet-ring interface (EX Series Switch Owner Node)

```
user@host> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg101

Interface      Forward State  RPL End  Signal Failure  Admin State

ge-0/0/3.0     discarding     Yes        Clear          ready
ge-0/0/9.0     forwarding     No         Clear          ready
```

show protection-group ethernet-ring interface (Owner Node MX Series Router)

```
user@host> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg101

Interface  Control Channel  Direction  Forward State  RPL End  SF     Admin State
ge-1/2/0   ge-1/2/0.100     east       forwarding     No       Clear  IFF ready
ge-1/2/2   ge-1/2/2.100     west       forwarding     No       Clear  IFF ready
```

show protection-group ethernet-ring interface detail (Owner Node MX Series Router)

```
user@host> show protection-group ethernet-ring interface detail
Ethernet ring port parameters for protection group pg101

Interface name                : ge-1/2/0
Control channel name          : ge-1/2/0.100
```

```
Interface direction            : east
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready

Interface name                 : ge-1/2/2
Control channel name           : ge-1/2/2.100
Interface direction            : west
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready
```

show protection-group ethernet-ring interface (EX Series Switch Ring Node)

```
user@host> show protection-group ethernet-ring interface
Ethernet ring port parameters for protection group pg102

Ethernet ring port parameters for protection group pg101

Interface      Forward State  RPL End  Signal Failure  Admin State

ge-0/0/3.0     discarding     Yes        Clear          ready
ge-0/0/9.0     forwarding     No         Clear          ready
```

show protection-group ethernet-ring interface detail (ACX Series and MX Series)

```
user@host> show protection-group ethernet-ring interface detail
Ethernet ring port parameters for protection group Erp_1

Interface name                 : xe-0/0/0
Control channel name           : xe-0/0/0.1
Interface direction            : east
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready

Interface name                 : et-0/0/48
Control channel name           : et-0/0/48.1
Interface direction            : west
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready
```

show protection-group ethernet-ring interface detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring interface detail
Ethernet ring port parameters for protection group pg1001

Interface name                 : ge-0/0/14
Control channel name           : ge-0/0/14.0
Interface direction            : east
Ring Protection Link End       : No
Signal Failure                 : Clear
Forward State                  : forwarding
Interface Admin State          : IFF ready
```

```
Interface name                : ge-0/0/18
Control channel name          : ge-0/0/18.0
Interface direction           : west
Ring Protection Link End      : No
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready
```

show protection-group ethernet-ring interface detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring interface detail
Ethernet ring port parameters for protection group pg1001

Interface name                : ge-0/0/42
Control channel name          : ge-0/0/42.0
Interface direction           : east
Ring Protection Link End      : Yes
Signal Failure                : Clear
Forward State                 : discarding
Interface Admin State         : IFF ready

Interface name                : ge-0/0/38
Control channel name          : ge-0/0/38.0
Interface direction           : west
Ring Protection Link End      : No
Signal Failure                : Clear
Forward State                 : forwarding
Interface Admin State         : IFF ready
```

show protection-group ethernet-ring node-state

Syntax **show protection-group ethernet-ring node-state**

Release Information Command introduced in Junos OS Release 9.4 for MX Series routers.
Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 12.3X54 for ACX Series routers.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description Display the status of the Automatic Protection Switching (APS) nodes on an Ethernet ring.

Options This command has no options.

Required Privilege Level view

Related Documentation

- show protection-group ethernet-ring data-channel on page 1444
- show protection-group ethernet-ring aps on page 1434
- show protection-group ethernet-ring interface on page 1447
- show protection-group ethernet-ring statistics on page 1456
- show protection-group ethernet-ring vlan on page 1462

List of Sample Output

show protection-group ethernet-ring node-state (MX Series Router - RPL Owner Node, Normal Operation) on page 1453
show protection-group ethernet-ring node-state (MX Series Router - Normal Ring Node, Normal Operation) on page 1453
show protection-group ethernet-ring node-state (MX Series Router - RPL Owner Node, Remote Failure Condition) on page 1453
show protection-group ethernet-ring node-state detail (ACX Series and MX Series Router) on page 1453
show protection-group ethernet-ring node-state detail (MX Series Router - RPL Owner Node, Normal Operation) on page 1454
show protection-group ethernet-ring node-state detail (MX Series Router with WTR Timer) on page 1454
show protection-group ethernet-ring node-state detail (MX Series Router with WTB Timer) on page 1454
show protection-group ethernet-ring node-state detail (EX2300 and EX3400 Switches) on page 1455

Output Fields Table 174 on page 1452 lists the output fields for the **show protection-group ethernet-ring node-state** command. Output fields are listed in the approximate order in which they appear.

Table 174: show protection-group ethernet-ring node-state Output Fields

Field Name	Field Description
Ring Name/Ethernet Ring	Name configured for the Ethernet ring.
APS State	State of the Ethernet ring APS. • **idle**—Indicates that the ring is working in normal condition and there is no active or pending protection-switching request in the ring. When the ring is in idle state, it is blocked at the RPL link. • **protected**—Indicates that there is a protection switch on the ring because of a signal failure condition on the ring link. • **MS**—Indicates that the manual switch command is active in the ring. • **FS**—Indicates that the forced switch command is active in the ring. • **pending**—Indicates that the ring is in pending state.
Event	Events on the ring. • **NR-RB**—Indicates that there is no APS request and the ring link is blocked on the ring owner node. • **NR**—Indicates that there is no APS request pending in the ring. • **local SF**—Indicates that there is signal failure on one or both of the ring links of the node. • **remote SF**—Indicates that there is signal failure on one or more ring links of any other node of the ring. • **local FS**—Indicates that there is a forced switched command active on one or both of the ring links of the node. • **remote FS**—Indicates that there is a forced switch command active on one or more ring links of any other node of the ring. • **local MS**—Indicates that there is a manual switch command active on one of the ring links of the node. • **remote MS**—Indicates that there is a manual switch command active on one or more ring links of any other node of the ring. • **WTR running**—Indicates that the wait to restore timer is running on the RPL owner. • **WTB running**—Indicates that the wait to block timer is running on the RPL owner.
RPL Owner / Ring Protection Link Owner	Whether this node is the ring owner: **Yes** or **No**.
WTR Timer / Restore Timer	Restoration timer: **running** or **disabled**.
WTB Timer / Wait to block timer	Wait to block timer: **running** or **disabled**. NOTE: The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.

Table 174: show protection-group ethernet-ring node-state Output Fields (continued)

Field Name	Field Description
Wait to block timer (WTB Timer)	Wait to block interval. NOTE: The Wait To Block Timer (WTB) is always disabled on EX2300 and EX3400 switches because it is not supported in ERPSv1. Any configuration you make to the WTB setting has no effect. The output from the CLI command 'show protection-group ethernet-ring node-state detail' lists a WTB setting but that setting has no effect.
Guard Timer	Guard timer: **running** or **disabled**.
Op State / Operational State	State of the node: **Operational** or **any internal wait state.**.

Sample Output

show protection-group ethernet-ring node-state (MX Series Router - RPL Owner Node, Normal Operation)

```
user@host> show protection-group ethernet-ring node-state
Ethernet ring   APS State  Event       RPL Owner  WTR Timer  WTB Timer  Guard
Timer  Operation state
pg101           idle       NR-RB       Yes        disabled   disabled   disabled
     operational
pg102            idle      NR-RB       No         disabled   disabled   disabled
     operational
```

show protection-group ethernet-ring node-state (MX Series Router - Normal Ring Node, Normal Operation)

```
user@host> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         RPL Owner
pg102            idle         NR-RB         No

WTR Timer  WTB Timer   Guard Timer  Operation state
disabled   disabled    disabled      operational
```

show protection-group ethernet-ring node-state (MX Series Router - RPL Owner Node, Remote Failure Condition)

```
user@host> show protection-group ethernet-ring node-state
Ethernet ring    APS State    Event         RPL Owner
pg101            protected    remote SF         Yes

WTR Timer   WTB Timer Guard Timer  Operation state
disabled    disabled   disabled     operational
```

show protection-group ethernet-ring node-state detail (ACX Series and MX Series Router)

```
user@host> show protection-group ethernet-ring node-state detail
Ethernet-Ring name              : Erp_1
APS State                       : idle
Event                           : NR-RB
Ring Protection Link Owner      : No
Wait to Restore Timer           : disabled
Wait to Block Timer             : disabled
```

```
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (MX Series Router - RPL Owner Node, Normal Operation)

```
user@host> show protection-group ethernet-ring node-state detail
Ethernet-Ring name             : pg101
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational

Ethernet-Ring name             : pg102
APS State                      : idle
Event                          : NR-RB
Ring Protection Link Owner     : No
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (MX Series Router with WTR Timer)

```
user@host> show protection-group ethernet-ring node-state detail
Ethernet-Ring name             : pg_major
APS State                      : pending
Event                          : WTR running
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : running (time to expire: 269 sec)
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational

Ethernet-Ring name             : pg_subring
APS State                      : pending
Event                          : NR
Ring Protection Link Owner     : No
Wait to Restore Timer          : disabled
Wait to Block Timer            : disabled
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (MX Series Router with WTB Timer)

```
user@host> show protection-group ethernet-ring node-state detail
Ethernet-Ring name             : Pg-2
APS State                      : pending
Event                          : WTB running
Ring Protection Link Owner     : Yes
Wait to Restore Timer          : disabled
Wait to Block Timer            : running (time to expire: 2 sec)
Guard Timer                    : disabled
Operation state                : operational
```

show protection-group ethernet-ring node-state detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring node-state detail
Ethernet-Ring name              : pg1001
APS State                       : idle
Event                           : NR-RB
Ring Protection Link Owner      : Yes
Wait to Restore Timer           : disabled
Wait to Block Timer             : disabled    <-field not supported. Always
disabled.
Guard Timer                     : disabled
Operation state                 : operational
```

show protection-group ethernet-ring statistics

Syntax **show protection-group ethernet-ring statistics group-name** ***group-name***

<brief | detail>

Release Information Command introduced in Junos OS Release 9.4.
Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 12.3X54 for ACX Series routers.

Description Display statistics regarding Automatic Protection Switching (APS) protection groups on an Ethernet ring.

Options **group-name**—Display statistics for the protection group. If you omit this option, protection group statistics for all configured groups are displayed.

brief—Display brief statistics for the protection group.

detail—Display detailed statistics for the protection group.

Required Privilege Level view

Related Documentation

- show protection-group ethernet-ring data-channel on page 1444
- show protection-group ethernet-ring aps on page 1434
- show protection-group ethernet-ring node-state on page 1451
- show protection-group ethernet-ring interface on page 1447
- show protection-group ethernet-ring vlan on page 1462

List of Sample Output

Output Fields Table 175 on page 1457 lists the output fields for the **show protection-group ethernet-ring statistics** command.

Table 175: show protection-group ethernet-ring statistics Output Fields

Field Name	Field Description
Ethernet Ring Statistics for PG	Name of the protection group for which statistics are displayed.
RAPS event sent	Number of times Ring Automatic Protection Switching (RAPS) message transmission event occurred locally. This field is applicable only to MX Series routers.
RAPS event received	Number of RAPS messages received and processed by ERP state-machine and which resulted in state transition. This field is applicable only to MX Series routers.
Local SF	Number of times a signal failure has occurred locally.
Remote SF	Number of times a signal failure has occurred anywhere else on the ring.
NR event	Number of times a No Request event has occurred on the ring. This field is applicable only to EX Series switches.
NR event sent	Number of times a No Request event has occurred locally. This field is applicable only to MX Series routers.
NR event received	Number of times a No Request event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.
NR-RB event	Number of times a No Request, Ring Blocked event has occurred on the ring. This field is applicable only to EX Series switches.
NR-RB event sent	Number of times a No Request, Ring Blocked event has occurred locally. This field is applicable only to MX Series routers.
NR-RB event received	Number of times a No Request, Ring Blocked event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.
Flush event sent	Number of times flush-event RAPS message transmission event occurred locally. This field is applicable only to MX Series routers.
Flush event received	Number of flush-event RAPS messages received and processed by the ring instance control process. This field is applicable only to MX Series routers.
Local FS event sent	Number of times a forced switch event has occurred locally. This field is applicable only to MX Series routers.
Remote FS event received	Number of times a forced switch event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.
Local MS event sent	Number of times a manual switch event has occurred locally. This field is applicable only to MX Series routers.

Table 175: show protection-group ethernet-ring statistics Output Fields (continued)

Field Name	Field Description
Remote MS event received	Number of times a manual switch event has occurred anywhere else on the ring. This field is applicable only to MX Series routers.

Table 176 on page 1458 lists the output fields for the **show protection-group ethernet-ring statistics** command when the **detail** option is used. These fields are valid only for MX Series routers.

Table 176: show protection-group ethernet-ring statistics detail Output Fields (for MX Series Routers)

Field Name	Field Description
Total number of FDB flush	Number of times forwarding database (FDB) flush has happened for the ring instance.
Flush-logic triggered flush	Number of times FDB flush has happened because of flush-logic based on node ID and Blocked Port Reference (BPR).
Remote RAPS PDU received	Number of valid RAPS PDU messages received. This counter counts only RAPS messages generated by other devices on the ring.
Remote RAPS dropped due to guard-timer	Number of RAPS messages dropped by the device because the guard timer is running.
Invalid remote RAPS PDU dropped	Number of RAPS messages dropped by the device because the messages are invalid.
RAPS dropped due to miscellaneous errors	Number of RAPS messages dropped because of any other reason. For example, messages dropped because of unsupported functionality.
Local received RAPS PDU dropped	Number of self-generated RAPS messages received and dropped.

Sample Output

show protection-group ethernet-ring statistics (EX Series Switch)

```
user@switch> show protection-group ethernet-ring statistics
Ring Name Local SF Remote SF NR Event NR-RB Event
erp1       2       1       2        3
```

show protection-group ethernet-ring statistics (MX Series Router)

```
user@host> show protection-group ethernet-ring statistics
Ethernet Ring statistics for PG Pg-1
RAPS event sent                  : 1
RAPS event received              : 1152
Local SF happened:               : 0
Remote SF happened:              : 428
```

```
NR event sent:                  : 1
NR event received:              : 133
NR-RB event sent:               : 0
NR-RB event received:           : 591
Flush event sent                : 0
Flush event received:           : 0
Local FS event sent:            : 0
Remote FS event received:       : 0
Local MS event sent:            : 0
Remote MS event received:       : 0
```

show protection-group ethernet-ring statistics detail (Specific Group)(MX Series Router)

```
user@host> show protection-group ethernet-ring statistics detail
Ethernet Ring statistics for PG Pg-1
RAPS event sent                          : 1
RAPS event received                      : 0
Local SF happened                        : 0
Remote SF happened                       : 0
NR event sent                            : 1
NR event received                        : 0
NR-RB event sent                         : 0
NR-RB event received                     : 0
Flush event sent                         : 0
Flush event received                     : 0
Local FS event sent                      : 0
Remote FS event received                 : 0
Local MS event sent                      : 0
Remote MS event received                 : 0
Total number of FDB flush                : 0
Flush-logic triggered flush              : 0
Remote raps PDU received                 : 0
Remote raps dropped due to guard-timer   : 0
Invalid remote raps PDU dropped          : 0
Raps dropped due to miscellaneous errors : 0
Local received raps PDU dropped          : 0
```

show protection-group ethernet-ring statistics (Owner Node, Failure Condition on ACX and MX Router)

```
user@host> show protection-group ethernet-ring statistics group-name pg101
Ethernet Ring statistics for PG pg101
RAPS sent                       : 1
RAPS received                   : 0
Local SF happened:              : 0
Remote SF happened:             : 0
NR event happened:              : 0
NR-RB event happened:           : 1
NR event sent:                  : 0
NR event received:              : 0
NR-RB event sent:               : 1
NR-RB event received:           : 0
Flush event sent                : 0
Flush event received:           : 0
Local FS event sent:            : 0
Remote FS event received:       : 0
Local MS event sent:            : 0
Remote MS event received:       : 0
```

show protection-group ethernet-ring statisitics (Ring Node, Failure Condition on ACX and MX Router)

```
user@host> show protection-group ethernet-ring statistics group-name pg102
Ethernet Ring statistics for PG pg102
RAPS sent                       : 1
RAPS received                   : 0
Local SF happened:              : 0
Remote SF happened:             : 0
NR event happened:              : 0
NR-RB event happened:           : 1
NR event sent:                  : 0
NR event received:              : 0
NR-RB event sent:               : 1
NR-RB event received:           : 0
Flush event sent                : 0
Flush event received:           : 0
Local FS event sent:            : 0
Remote FS event received:       : 0
Local MS event sent:            : 0
Remote MS event received:       : 0
```

show protection-group ethernet-ring statistics detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring statistics detail
Ethernet Ring statistics for PG pg1001
RAPS event sent                          : 1
RAPS event received                      : 1
Local SF happened                        : 0
Remote SF happened                       : 0
NR event sent                            : 1
NR event received                        : 0
NR-RB event sent                         : 0
NR-RB event received                     : 1
Flush event sent                         : 0
Flush event received                     : 0
Local FS event sent                      : 0
Remote FS event received                 : 0
Local MS event sent                      : 0
Remote MS event received                 : 0
Total number of FDB flush                : 0
Flush-logic triggered flush              : 0
Remote raps PDU received                 : 145
Remote raps dropped due to guard-timer   : 0
Invalid remote raps PDU dropped          : 0
Raps dropped due to miscellaneous errors : 0
Local received raps PDU dropped          : 0
```

show protection-group ethernet-ring statistics detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring statistics detail
Ethernet Ring statistics for PG pg1001
RAPS event sent                          : 2
RAPS event received                      : 0
Local SF happened                        : 0
Remote SF happened                       : 0
NR event sent                            : 1
NR event received                        : 0
NR-RB event sent                         : 1
NR-RB event received                     : 0
Flush event sent                         : 0
```

```
Flush event received                     : 0
Total number of FDB flush                : 0
Remote raps PDU received                 : 211
Remote raps dropped due to guard-timer   : 0
Invalid remote raps PDU dropped          : 0
Raps dropped due to miscellaneous errors : 0
Local received raps PDU dropped          : 91
```

show protection-group ethernet-ring vlan

Syntax

show protection-group ethernet-ring vlan
<brief | detail>
<group-name *group-name*>

Release Information

Command introduced in Junos OS Release 10.2.
Command introduced in Junos OS Release 18.1 for EX2300 and EX3400 switches.

Description

On MX Series routers, display all data channel logical interfaces and the VLAN IDs controlled by a ring instance data channel.

Options

brief | detail—(Optional) Display the specified level of output.

group-name—(Optional) Protection group for which to display details such as data channel interfaces, vlan, and bridge-domain. If you omit this optional field, details for all configured protection groups will be displayed.

Required Privilege Level

view

Related Documentation

- show protection-group ethernet-ring aps on page 1434
- show protection-group ethernet-ring data-channel on page 1444
- show protection-group ethernet-ring interface on page 1447
- show protection-group ethernet-ring node-state on page 1451
- show protection-group ethernet-ring statistics on page 1456

List of Sample Output

show protection-group ethernet-ring vlan on page 1463
show protection-group ethernet-ring vlan brief on page 1463
show protection-group ethernet-ring vlan detail on page 1464
show protection-group ethernet-ring vlan group-name vkm01 on page 1465
show protection-group ethernet-ring vlan detail (EX2300 and EX3400 Switches) on page 1465

Output Fields

Table 177 on page 1462 lists the output fields for the **show protection-group ethernet-ring vlan** command. Output fields are listed in the approximate order in which they appear.

Table 177: show protection-group ethernet-ring vlan Output Fields

Field Name	Field Description
Interface	Name of the interface configured for the Ethernet protection ring.
Vlan	Name of the VLAN associated with the interface configured for the Ethernet protection ring.

Table 177: show protection-group ethernet-ring vlan Output Fields (continued)

Field Name	Field Description
STP index	The Spanning Tree Protocol (STP) index number used by each interface in an Ethernet ring. The STP index controls the forwarding behavior for a set of VLANs on a data channel on an Ethernet ring port. For multiple Ethernet ring instances on an physical ring port, there are multiple STP index numbers. Different ring instances will have different STP index numbers and may have different forwarding behavior.
Bridge Domain	Name of the bridge domain that is associated with the VLAN configured for the Ethernet protection ring.

Sample Output

show protection-group ethernet-ring vlan

```
user@host> show protection-group ethernet-ring vlan
Ethernet ring IFBD parameters for protection group vkm01

Interface     Vlan     STP Index   Bridge Domain
xe-5/0/2      1        78          default-switch/bd1
xe-2/2/0      1        79          default-switch/bd1
xe-5/0/2      2        78          default-switch/bd2
xe-2/2/0      2        79          default-switch/bd2
xe-5/0/2      3        78          default-switch/bd3
xe-2/2/0      3        79          default-switch/bd3
xe-5/0/2      4        78          default-switch/bd4
xe-2/2/0      4        79          default-switch/bd4
xe-5/0/2      5        78          default-switch/bd5
xe-2/2/0      5        79          default-switch/bd5
xe-5/0/2      6        78          default-switch/bd6
xe-2/2/0      6        79          default-switch/bd6
xe-5/0/2      7        78          default-switch/bd7
xe-2/2/0      7        79          default-switch/bd7
xe-5/0/2      8        78          default-switch/bd8
xe-2/2/0      8        79          default-switch/bd8
xe-5/0/2      9        78          default-switch/bd9
xe-2/2/0      9        79          default-switch/bd9
xe-5/0/2      10       78          default-switch/bd10
xe-2/2/0      10       79          default-switch/bd10
xe-5/0/2      11       78          default-switch/bd11
xe-2/2/0      11       79          default-switch/bd11
xe-5/0/2      12       78          default-switch/bd12
xe-2/2/0      12       79          default-switch/bd12
xe-5/0/2      13       78          default-switch/bd13
xe-2/2/0      13       79          default-switch/bd13
xe-5/0/2      14       78          default-switch/bd14
xe-2/2/0      14       79          default-switch/bd14
xe-5/0/2      15       78          default-switch/bd15
xe-2/2/0      15       79          default-switch/bd15
```

show protection-group ethernet-ring vlan brief

```
user@host> show protection-group ethernet-ring vlan brief
```

```
Ethernet ring IFBD parameters for protection group vkm01

Interface    Vlan    STP Index   Bridge Domain
xe-5/0/2     1       78          default-switch/bd1
xe-2/2/0     1       79          default-switch/bd1
xe-5/0/2     2       78          default-switch/bd2
xe-2/2/0     2       79          default-switch/bd2
xe-5/0/2     3       78          default-switch/bd3
xe-2/2/0     3       79          default-switch/bd3
xe-5/0/2     4       78          default-switch/bd4
xe-2/2/0     4       79          default-switch/bd4
xe-5/0/2     5       78          default-switch/bd5
xe-2/2/0     5       79          default-switch/bd5
xe-5/0/2     6       78          default-switch/bd6
xe-2/2/0     6       79          default-switch/bd6
xe-5/0/2     7       78          default-switch/bd7
xe-2/2/0     7       79          default-switch/bd7
xe-5/0/2     8       78          default-switch/bd8
xe-2/2/0     8       79          default-switch/bd8
xe-5/0/2     9       78          default-switch/bd9
xe-2/2/0     9       79          default-switch/bd9
xe-5/0/2     10      78          default-switch/bd10
xe-2/2/0     10      79          default-switch/bd10
xe-5/0/2     11      78          default-switch/bd11
xe-2/2/0     11      79          default-switch/bd11
xe-5/0/2     12      78          default-switch/bd12
xe-2/2/0     12      79          default-switch/bd12
xe-5/0/2     13      78          default-switch/bd13
xe-2/2/0     13      79          default-switch/bd13
xe-5/0/2     14      78          default-switch/bd14
xe-2/2/0     14      79          default-switch/bd14
xe-5/0/2     15      78          default-switch/bd15
xe-2/2/0     15      79          default-switch/bd15
```

show protection-group ethernet-ring vlan detail

```
user@host> show protection-group ethernet-ring vlan detail
Ethernet ring IFBD parameters for protection group vkm01

Interface name                : xe-5/0/2
Vlan                          : 1
STP index                     : 78
Bridge Domain                 : default-switch/bd1

Interface name                : xe-2/2/0
Vlan                          : 1
STP index                     : 79
Bridge Domain                 : default-switch/bd1

Interface name                : xe-5/0/2
Vlan                          : 2
STP index                     : 78
Bridge Domain                 : default-switch/bd2

Interface name                : xe-2/2/0
Vlan                          : 2
STP index                     : 79
Bridge Domain                 : default-switch/bd2

Interface name                : xe-5/0/2
Vlan                          : 3
```

```
STP index                       : 78
Bridge Domain                   : default-switch/bd3
```

show protection-group ethernet-ring vlan group-name vkm01

```
user@host> show protection-group ethernet-ring vlan vkm01

Ethernet ring IFBD parameters for protection group vkm01

Interface    Vlan    STP Index   Bridge Domain
xe-5/0/2     16      80          default-switch/bd16
xe-2/2/0     16      81          default-switch/bd16
xe-5/0/2     17      80          default-switch/bd17
xe-2/2/0     17      81          default-switch/bd17
xe-5/0/2     18      80          default-switch/bd18
xe-2/2/0     18      81          default-switch/bd18
xe-5/0/2     19      80          default-switch/bd19
xe-2/2/0     19      81          default-switch/bd19
xe-5/0/2     20      80          default-switch/bd20
xe-2/2/0     20      81          default-switch/bd20
xe-5/0/2     21      80          default-switch/bd21
xe-2/2/0     21      81          default-switch/bd21
xe-5/0/2     22      80          default-switch/bd22
xe-2/2/0     22      81          default-switch/bd22
xe-5/0/2     23      80          default-switch/bd23
xe-2/2/0     23      81          default-switch/bd23
xe-5/0/2     24      80          default-switch/bd24
xe-2/2/0     24      81          default-switch/bd24
xe-5/0/2     25      80          default-switch/bd25
xe-2/2/0     25      81          default-switch/bd25
xe-5/0/2     26      80          default-switch/bd26
xe-2/2/0     26      81          default-switch/bd26
xe-5/0/2     27      80          default-switch/bd27
xe-2/2/0     27      81          default-switch/bd27
xe-5/0/2     28      80          default-switch/bd28
xe-2/2/0     28      81          default-switch/bd28
xe-5/0/2     29      80          default-switch/bd29
xe-2/2/0     29      81          default-switch/bd29
xe-5/0/2     30      80          default-switch/bd30
xe-2/2/0     30      81          default-switch/bd30
```

show protection-group ethernet-ring vlan detail (EX2300 and EX3400 Switches)

```
user@switch>show protection-group ethernet-ring vlan detail
Ethernet ring IFBD parameters for protection group pg1001

Interface name                  : ge-0/0/42
Vlan                            : 2001
STP index                       : 52
Bridge Domain                   : default-switch/vlan2001

Interface name                  : ge-0/0/38
Vlan                            : 2001
STP index                       : 53
Bridge Domain                   : default-switch/vlan2001
```

show redundant-trunk-group

Syntax **show redundant-trunk-group <group-name *group-name*>**

Release Information Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 13.2X50-D15 for the QFX Series.

Description Display information about redundant trunk groups.

Options **group-name *group-name***—Display information about the specified redundant trunk group.

Required Privilege Level view

Related Documentation
- Example: Configuring Redundant Trunk Links for Faster Recovery on EX Series Switches on page 619
- Example: Configuring Redundant Trunk Links for Faster Recovery on Devices with ELS Support on page 613
- Understanding Redundant Trunk Links (Legacy RTG Configuration) on page 610

List of Sample Output show redundant-trunk-group group-name Group1 on page 1467

Output Fields Table 178 on page 1466 lists the output fields for the **show redundant-trunk-group** command. Output fields are listed in the approximate order in which they appear.

Table 178: show redundant-trunk-group Output Fields

Field Name	Field Description
Group name	Name of the redundant trunk port group.
Interface	Name of an interface belonging to the trunk port group.
State	Operating state of the interface. • **Up** denotes the interface is up. • **Down** denotes the interface is down. • **Pri** denotes a primary interface. • **Act** denotes an active interface.
Time of last flap	Date and time at which the advertised link became unavailable, and then, available again.
Flap count	Total number of flaps since the last switch reboot.

Sample Output

show redundant-trunk-group group-name Group1

```
user@switch> show redundant-trunk-group group-name Group1

Group       Interface       State          Time of last flap       Flap
name                                                               Count

Group1      ge-0/0/45.0     UP/Pri/Act     Never                   0
            ge-0/0/47.0     UP             Never                   0
```

show security flow gate family

Syntax show security flow gate family (inet | inet6)

Release Information Command introduced in Junos OS Release 10.4.

Description Display filtered summary of information about existing gates, types of gates, and the maximum allowed number of gates.

Options
- inet—Displays IPv4 information.
- inet6—Displays IPv6 gate information.

Required Privilege Level view

Related Documentation
- *show security flow gate*

Output Fields Table 179 on page 1468 lists the output fields for the **show security flow gate family** command. Output fields are listed in the approximate order in which they appear.

Table 179: show security flow gate family Output Fields

Field Name	Field Description
Valid gates	Number of valid gates.
Pending gates	Number of pending gates.
Invalidated gates	Number of invalid gates.
Gates in other states	Number of gates in other states.
Total gates	Total number of gates.

Sample Output

```
user@host> show security flow gate family inet6
Hole: 2001:13::8-0-0->2001:12::8-33135-33135

  Translated: ::/0->::/0

  Protocol: tcp

  Application: FTP ALG/79

  Age: 24 seconds
```

```
  Flags: 0x8080

  Zone: zserver

  Reference count: 1

  Resource: 1-2-2


Valid gates: 1

Pending gates: 0

Invalidated gates: 0

Gates in other states: 0

Total gates: 1
```

```
user@host> show security flow gate family inet6 destination-prefix 2001:12::8 or source-prefix
Hole: 2001:13::8-0-0->2001:12::8-33135-33135

  Translated: ::/0->::/0

  Protocol: tcp

  Application: FTP ALG/79

  Age: 26 seconds

  Flags: 0x8080

  Zone: zserver

  Reference count: 1

  Resource: 1-2-2


Valid gates: 1

Pending gates: 0

Invalidated gates: 0

Gates in other states: 0

Total gates: 1
```

show security flow ip-action

Syntax **show security flow ip-action** [<filter>] [summary family (inet | inet6)]

Release Information Command introduced in Junos OS Release 10.1. Logical systems option added in Junos OS Release 11.2 . Summary option introduced in Junos OS Release 12.1.

Description Display the current IP-action settings, based on filtered options, for IP sessions running on the device.

Options

- *filter*—Filter the display based on the specified criteria.

  The following filters display those sessions that match the criteria specified by the filter. Refer to the sample output for filtered output examples.

  all | [*filter*]—All active sessions on the device.

  destination-port ***destination-port***—Destination port number of the traffic. Range is 1 through 65,535.

  destination-prefix ***destination-prefix***—Destination IP prefix or address.

  family (inet | inet6) [*filter*]—IPv4 traffic or IPv6-NATPT traffic and filtered options.

  logical-system ***logical-system-name*** **| all** [*filter*]—Specified logical system or all logical systems.

  protocol ***protocol-name*** **|** ***protocol-number*** [*filter*]—Protocol name or number and filtered options.

  - **ah** or **51**
  - **egp** or **8**
  - **esp** or **50**
  - **gre** or **47**
  - **icmp** or **1**
  - **icmp6** or **58**
  - **ipip** or **4**
  - **ospf** or **89**
  - **pim** or **103**
  - **rsvp** or **46**
  - **sctp** or **132**
  - **tcp** or **6**
  - **udp** or **17**

  root-logical-system [*filter*]—Default logical system information and filtered options.

source-port *source-port*—Source port number of the traffic. Range is 1 through 65,535.

source-prefix *source-prefix*—Source IP prefix or address of the traffic.

- **summary** —Summary information about IP-action entries.

  family—Display summary of IP-action entries by family. This option is used to filter the output.

  - **inet**—Display summary of IPv4 entries.
  - **inet6**—Display summary of IPv6 entries.

Required Privilege Level view

Related Documentation

- *Understanding Traffic Processing on Security Devices*
- clear security flow ip-action on page 1180
- *clear security flow session destination-port*

List of Sample Output

show security flow ip-action on page 1472
show security flow ip-action destination-port on page 1473
show security flow ip-action destination-prefix on page 1474
show security flow ip-action family inet protocol on page 1474
show security flow ip-action family inet logical-system all on page 1475
show security flow ip-action source-prefix on page 1476
show security flow ip-action summary on page 1477
show security flow ip-action summary family inet on page 1477
show security flow ip-action summary family inet6 on page 1477

Output Fields Table 180 on page 1471 lists the output fields for the **show security flow ip-action** command. Output fields are listed in the approximate order in which they appear.

Table 180: show security flow ip-action Output Fields

Field Name	Field Description
Src-Addr	Source address of outbound IP traffic.
Src-Port	Source port number of outbound IP traffic.
Dst-Addr	Destination address of inbound IP traffic.
Dst-Port/Proto	Destination port number and protocol type of inbound IP traffic.
Timeout (sec)	Configured timeouts and time remaining for an IP session.
Zone	Security zone associated with an IP session.
Action	Configured action type, for example, block, close, and notify.

Table 180: show security flow ip-action Output Fields (continued)

Field Name	Field Description
State	The active mode and passive mode describe the states of the **ip-action** entry.
IPv4 action count	The total number of IPv4 entries.
IPv6 action count	The total number of IPv6 entries.

Sample Output

show security flow ip-action

```
user@host> show security flow ip-action
Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
 203.0.113.1       *       203.0.113.4         21/tcp         293/300      *
      close        Passive
 IPv4 action count: 1 on FPC0.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *      203.0.113.4          21/tcp         293/300      *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *      203.0.113.4          21/tcp         293/300      *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC3

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *       203.0.113.4          21/tcp         293/300      *
  close        Passive
 IPv4 action count: 1 on FPC1.PIC0

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *      203.0.113.4          21/tcp         293/300       *
    close        Passive
 IPv4 action count: 1 on FPC1.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *    203.0.113.4          21/tcp         292/300      *
   close        Passive
 IPv4 action count: 1 on FPC1.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
 203.0.113.1         *    203.0.113.4          21/tcp         292/300      *
   close        Active
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
 IPv6 action count: 0 on FPC0.PIC1
```

```
 IPv6 action count: 0 on FPC0.PIC2
 IPv6 action count: 0 on FPC0.PIC3
 IPv6 action count: 0 on FPC1.PIC0
 IPv6 action count: 0 on FPC1.PIC1
 IPv6 action count: 0 on FPC1.PIC2
 IPv6 action count: 0 on FPC1.PIC3
 IPv6 action count: Active mode 0 on all PICs
```

show security flow ip-action destination-port

```
user@host> show security flow ip-action destination-port 21

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp         274/300      *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp         274/300      *
 close        Passive
 IPv4 action count: 1 on FPC0.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp         274/300      *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC3

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4        21/tcp         274/300      *
  close        Passive
 IPv4 action count: 1 on FPC1.PIC0

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp         274/300      *
  close        Passive
 IPv4 action count: 1 on FPC1.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1        *        203.0.113.4         21/tcp         274/300      *
  close        Passive
 IPv4 action count: 1 on FPC1.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp         273/300      *
  close        Active
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
 IPv6 action count: 0 on FPC0.PIC1
 IPv6 action count: 0 on FPC0.PIC2
 IPv6 action count: 0 on FPC0.PIC3
 IPv6 action count: 0 on FPC1.PIC0
 IPv6 action count: 0 on FPC1.PIC1
 IPv6 action count: 0 on FPC1.PIC2
```

```
 IPv6 action count: 0 on FPC1.PIC3
 IPv6 action count: Active mode 0 on all PICs
```

show security flow ip-action destination-prefix

```
user@host>  show security flow ip-action destination-prefix 203.0.113.4/8

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp          245/300      *
 close        Passive
 IPv4 action count: 1 on FPC0.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp          245/300      *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp          245/300      *
 close        Passive
 IPv4 action count: 1 on FPC0.PIC3

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
192.0.2.3          *        203.0.113.4         21/tcp          245/300      *
 close        Passive
 IPv4 action count: 1 on FPC1.PIC0

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
192.0.2.3          *        203.0.113.4         21/tcp          245/300      *
 close        Passive
 IPv4 action count: 1 on FPC1.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp          245/300      *
  close        Passive
 IPv4 action count: 1 on FPC1.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         21/tcp          245/300      *
 close        Active
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
```

show security flow ip-action family inet protocol

```
user@host>  show security flow ip-action family inet protocoludp
Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         69/udp          287/300      *
        close        Passive
 IPv4 action count: 1 on FPC0.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
```

```
Action       State
203.0.113.1         *        203.0.113.4         69/udp         287/300     *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         69/udp         287/300      *
    close        Passive
 IPv4 action count: 1 on FPC0.PIC3

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         69/udp         287/300      *
     close        Active
 IPv4 action count: 1 on FPC1.PIC0

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         69/udp         287/300      *
     close        Passive
 IPv4 action count: 1 on FPC1.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1        *        203.0.113.4         69/udp         287/300      *
    close        Passive
 IPv4 action count: 1 on FPC1.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        203.0.113.4         69/udp         287/300      *
    close        Passive
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
```

show security flow ip-action family inet logical-system all

```
user@host> show security flow ip-action family inet logical-system all

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *        203.0.113.4         69/udp         267/300      *
        close        Passive           root-logical-system
 IPv4 action count: 1 on FPC0.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *        203.0.113.4         69/udp         267/300      *
 close        Passive           root-logical-system
 IPv4 action count: 1 on FPC0.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *       203.0.113.4         69/udp         267/300      *
  close        Passive           root-logical-system
 IPv4 action count: 1 on FPC0.PIC3

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *        203.0.113.4         69/udp         267/300      *
```

```
  close        Active            root-logical-system
 IPv4 action count: 1 on FPC1.PIC0

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *        203.0.113.4         69/udp         267/300      *
  close        Passive           root-logical-system
 IPv4 action count: 1 on FPC1.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *        203.0.113.4        69/udp         266/300      *
  close        Passive           root-logical-system
 IPv4 action count: 1 on FPC1.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State             Logical-System
203.0.113.1         *        203.0.113.4        69/udp         266/300      *
  close        Passive           root-logical-system
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
```

show security flow ip-action source-prefix

```
user@host>  show security flow ip-action source-prefix 192.0.2.3/8

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4         69/udp         244/300      *
  close        Passive
 IPv4 action count: 1 on FPC0.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4         69/udp         244/300      *
   close        Passive
 IPv4 action count: 1 on FPC0.PIC2

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4        69/udp         244/300      *
    close        Passive
 IPv4 action count: 1 on FPC0.PIC3

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4         69/udp         244/300      *
     close        Active
 IPv4 action count: 1 on FPC1.PIC0

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4         69/udp         244/300      *
    close        Passive
 IPv4 action count: 1 on FPC1.PIC1

Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4         69/udp         244/300      *
     close        Passiveo
 IPv4 action count: 1 on FPC1.PIC2
```

```
Src-Addr        Src-Port Dst-Addr        Dst-Port/Proto Timeout(sec) Zone
Action       State
203.0.113.1         *        192.0.2.4         69/udp         244/300      *
     close        Passive
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
```

show security flow ip-action summary

```
user@host> show security flow ip-action summary

 IPv4 action count: 1 on FPC0.PIC1
 IPv4 action count: 1 on FPC0.PIC2
 IPv4 action count: 1 on FPC0.PIC3
 IPv4 action count: 1 on FPC1.PIC0
 IPv4 action count: 1 on FPC1.PIC1
 IPv4 action count: 1 on FPC1.PIC2
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
 IPv6 action count: 0 on FPC0.PIC1
 IPv6 action count: 0 on FPC0.PIC2
 IPv6 action count: 0 on FPC0.PIC3
 IPv6 action count: 0 on FPC1.PIC0
 IPv6 action count: 0 on FPC1.PIC1
 IPv6 action count: 0 on FPC1.PIC2
 IPv6 action count: 0 on FPC1.PIC3
 IPv6 action count: Active mode 0 on all PICs
```

show security flow ip-action summary family inet

```
user@host> show security flow ip-action summary inet

 IPv4 action count: 1 on FPC0.PIC1
 IPv4 action count: 1 on FPC0.PIC2
 IPv4 action count: 1 on FPC0.PIC3
 IPv4 action count: 1 on FPC1.PIC0
 IPv4 action count: 1 on FPC1.PIC1
 IPv4 action count: 1 on FPC1.PIC2
 IPv4 action count: 1 on FPC1.PIC3
 IPv4 action count: Active mode 1 on all PICs
```

show security flow ip-action summary family inet6

```
user@host> show security flow ip-action summary family inet6

 IPv6 action count: 1 on FPC0.PIC1
 IPv6 action count: 1 on FPC0.PIC2
 IPv6 action count: 1 on FPC0.PIC3
 IPv6 action count: 1 on FPC1.PIC0
 IPv6 action count: 1 on FPC1.PIC1
 IPv6 action count: 1 on FPC1.PIC2
 IPv6 action count: 1 on FPC1.PIC3
 IPv6 action count: Active mode 1 on all PICs
```

show security flow session family

Syntax show security flow session family (inet | inet6)
[brief | extensive | summary]

Release Information Command introduced in Junos OS Release 10.2.

Description Display filtered summary of information about existing sessions, including types of sessions, active and failed sessions, and the maximum allowed number of sessions.

Options
- **inet**—Display details summary of IPv4 sessions.
- **inet6**—Display details summary of IPv6 sessions.
- brief | extensive | summary–Display the specified level of output.

Required Privilege Level view

Related Documentation
- *Understanding Traffic Processing on Security Devices*
- clear security flow session family on page 1182

List of Sample Output
show security flow session family inet on page 1479
show security flow session family inet brief on page 1480
show security flow session family inet extensive on page 1480
show security flow session family inet summary on page 1482

Output Fields Table 181 on page 1478 lists the output fields for the **show security flow session family** command. Output fields are listed in the approximate order in which they appear.

Table 181: show security flow session family Output Fields

Field Name	Field Description
Session ID	Number that identifies the session. Use this ID to get more information about the session.
Policy name	Policy that permitted the traffic.
Timeout	Idle timeout after which the session expires.
In	Incoming flow (source and destination IP addresses, application protocol, interface, session token, route, gateway, tunnel, port sequence, FIN sequence, FIN state, packets and bytes).
Out	Reverse flow (source and destination IP addresses, application protocol, interface, session token, route, gateway, tunnel, port sequence, FIN sequence, FIN state, packets and bytes).

Table 181: show security flow session family Output Fields (continued)

Field Name	Field Description
Total sessions	Total number of sessions.
Status	Session status.
Flag	Internal flag depicting the state of the session, used for debugging purposes.
Policy name	Name and ID of the policy that the first packet of the session matched.
Source NAT pool	The name of the source pool where NAT is used.
Application	Name of the application.
Maximum timeout	Maximum session timeout.
Current timeout	Remaining time for the session unless traffic exists in the session.
Session State	Session state.
Start time	Time when the session was created, offset from the system start time.
Unicast-sessions	Number of unicast sessions.
Multicast-sessions	Number of multicast sessions.
Failed-sessions	Number of failed sessions.
Sessions-in-use	Number of sessions in use. • Valid sessions • Pending sessions • Invalidated sessions • Sessions in other states
Maximum-sessions	Number of maximum sessions.

Sample Output

show security flow session family inet

```
root> show security flow session family inet
Flow Sessions on FPC10 PIC1:
Total sessions: 0

Flow Sessions on FPC10 PIC2:

Session ID: 420000107, Policy name: default-policy-00/2, Timeout: 4, Valid
  In: 203.0.113.0/3 --> 203.0.113.5/24;icmp, If: ge-7/1/0.0, Pkts: 1, Bytes: 84,
 CP Session ID: 420000202
```

```
  Out: 203.0.113.4/24 --> 203.0.113.6/24;icmp, If: .local..0, Pkts: 1, Bytes: 84,
 CP Session ID: 420000202
Total sessions: 1

Flow Sessions on FPC10 PIC3:

Session ID: 430000115, Policy name: default-policy-00/2, Timeout: 2, Valid
  In: 203.0.113.0/4 --> 203.0.113.5/24;icmp, If: ge-7/1/0.0, Pkts: 1, Bytes: 84,
 CP Session ID: 430000110
  Out: 203.0.113.5/24 --> 203.0.113.6/24;icmp, If: .local..0, Pkts: 1, Bytes: 84,
 CP Session ID: 430000110

Session ID: 430000117, Policy name: default-policy-00/2, Timeout: 4, Valid
  In: 203.0.113.0/4 --> 203.0.113.5/24;icmp, If: ge-7/1/0.0, Pkts: 1, Bytes: 84,
 CP Session ID: 430000111
  Out: 203.0.113.5/24 --> 203.0.113.6/24;icmp, If: .local..0, Pkts: 1, Bytes: 84,
 CP Session ID: 430000111
Total sessions: 2
```

show security flow session family inet brief

```
root> show security flow session family inet brief
Flow Sessions on FPC10 PIC1:
Total sessions: 0

Flow Sessions on FPC10 PIC2:

Session ID: 420000115, Policy name: default-policy-00/2, Timeout: 2, Valid
  In: 203.0.113.0/3 --> 203.0.113.5/24;icmp, If: ge-7/1/0.0, Pkts: 1, Bytes: 84,
 CP Session ID: 420000206
  Out: 203.0.113.4/24 --> 203.0.113.6/24;icmp, If: .local..0, Pkts: 1, Bytes: 84,
 CP Session ID: 420000206

Session ID: 420000117, Policy name: default-policy-00/2, Timeout: 2, Valid
  In: 203.0.113.0/4 --> 203.0.113.5/24;icmp, If: ge-7/1/0.0, Pkts: 1, Bytes: 84,
 CP Session ID: 420000207
  Out: 203.0.113.5/24 --> 203.0.113.6/24;icmp, If: .local..0, Pkts: 1, Bytes: 84,
 CP Session ID: 420000207
Total sessions: 2

Flow Sessions on FPC10 PIC3:

Session ID: 430000119, Policy name: default-policy-00/2, Timeout: 2, Valid
  In: 203.0.113.0/4 --> 203.0.113.5/24;icmp, If: ge-7/1/0.0, Pkts: 1, Bytes: 84,
 CP Session ID: 430000112
  Out: 203.0.113.5/24 --> 203.0.113.6/24;icmp, If: .local..0, Pkts: 1, Bytes: 84,
 CP Session ID: 430000112
Total sessions: 1
```

show security flow session family inet extensive

```
root> show security flow session family inet extensive
Flow Sessions on FPC10 PIC1:

Session ID: 410000111, Status: Normal
Flags: 0x80400040/0x0/0x2800023
Policy name: default-policy-00/2
Source NAT pool: Null
Dynamic application: junos:UNKNOWN,
Encryption:  Unknown
```

```
Application traffic control rule-set: INVALID, Rule: INVALID
Maximum timeout: 4, Current timeout: 4
Session State: Valid
Start time: 76455, Duration: 0
   In: 203.0.113.0/24 --> 203.0.113.1/24;icmp,
    Interface: ge-7/1/0.0,
    Session token: 0x6, Flag: 0xc0000021
    Route: 0xa0010, Gateway: 203.0.113.10, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 1, Bytes: 84
    CP Session ID: 410000242
   Out: 203.0.113.1/24 --> 203.0.113.10/4;icmp,
    Interface: .local..0,
    Session token: 0x2, Flag: 0x40000030
    Route: 0xfffb0006, Gateway: 203.0.113.1, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 1, Bytes: 84
    CP Session ID: 410000242
Total sessions: 1

Flow Sessions on FPC10 PIC2:

Session ID: 420000123, Status: Normal
Flags: 0x80400040/0x0/0x2800023
Policy name: default-policy-00/2
Source NAT pool: Null
Dynamic application: junos:UNKNOWN,
Encryption:  Unknown
Application traffic control rule-set: INVALID, Rule: INVALID
Maximum timeout: 4, Current timeout: 2
Session State: Valid
Start time: 76454, Duration: 2
   In: 203.0.113.10/24 --> 203.0.113.11/24;icmp,
    Interface: ge-7/1/0.0,
    Session token: 0x6, Flag: 0xc0000021
    Route: 0xa0010, Gateway: 20010, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 1, Bytes: 84
    CP Session ID: 420000210
   Out: 203.0.113.11/24 --> 203.0.113.12/24;icmp,
    Interface: .local..0,
    Session token: 0x2, Flag: 0x40000030
    Route: 0xfffb0006, Gateway: 203.0.113.1, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 1, Bytes: 84
    CP Session ID: 420000210
Total sessions: 1

Flow Sessions on FPC10 PIC3:

Session ID: 430000131, Status: Normal
Flags: 0x80400040/0x0/0x2800023
Policy name: default-policy-00/2
Source NAT pool: Null
Dynamic application: junos:UNKNOWN,
Encryption:  Unknown
Application traffic control rule-set: INVALID, Rule: INVALID
```

```
Maximum timeout: 4, Current timeout: 4
Session State: Valid
Start time: 76421, Duration: 1
   In: 203.0.113.10/24 --> 203.0.113.11/24;icmp,
    Interface: ge-7/1/0.0,
    Session token: 0x6, Flag: 0xc0000021
    Route: 0xa0010, Gateway: 203.0.113.10, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 1, Bytes: 84
    CP Session ID: 430000118
   Out: 203.0.113.12/24 --> 203.0.113.13/24;icmp,
    Interface: .local..0,
    Session token: 0x2, Flag: 0x40000030
    Route: 0xfffb0006, Gateway: 203.0.113.1, Tunnel: 0
    Port sequence: 0, FIN sequence: 0,
    FIN state: 0,
    Pkts: 1, Bytes: 84
    CP Session ID: 430000118
Total sessions: 1
```

show security flow session family inet summary

```
root> show security flow session family inet summary
Flow Sessions on FPC10 PIC1:

Valid sessions: 2
Pending sessions: 0
Invalidated sessions: 2
Sessions in other states: 0
Total sessions: 4

Flow Sessions on FPC10 PIC2:

Valid sessions: 2
Pending sessions: 0
Invalidated sessions: 2
Sessions in other states: 0
Total sessions: 4

Flow Sessions on FPC10 PIC3:

Valid sessions: 2
Pending sessions: 0
Invalidated sessions: 2
Sessions in other states: 0
Total sessions: 4
```

show security flow statistics

Syntax **show security flow statistics**
<node (*node-id* | all | local | primary)>

Release Information Command introduced in Junos OS Release 10.2. Fragmentation counters options introduced in Junos OS Release 15.1X49-90.

Description Display security flow statistics on a specific SPU. A flow is a stream of related packets that meet the same matching criteria and share the same characteristics.

A packet undergoes flow-based processing after packet-based filters and some screens have been applied to it. A System Processing Unit (SPU) processes the packets of a flow according to the security features and other services configured for the session. Flow-based packet processing treats related packets, or a stream of packets, in the same way. Packet treatment depends on characteristics that were established for the first packet of the packet stream.

The **show security flow statistics** command displays information for individual SPUs. For each SPU, the active sessions on the SPU, packets received, packets transmitted, packets forwarded/queued, packets copied, packets dropped, packet fragments received in a flow on the SPU, pre-fragmented packets generated, and post-fragmented packets generated are displayed in terms of numbers.

There are many conditions that can cause a packet to be dropped. Here are some of them:

- A screen module detects IP spoofing
- The IPSec Encapsulating Security Payload (ESP) or the Authentication Header (AH) authentication failed. For example, incoming NAT errors could cause this to happen.
- A packet matches more than one security policy that specifies user authentication. (Sometimes packets are looped through the system more than once. Each time a packet passes through the system, that packet must be permitted by a policy.)
- A time constraint setting expires. For example, multicast streams with a packet interval of more than 60 seconds would experience premature aging-out of flow sessions. (In most cases, you can configure higher time-out value to prevent packet drop.)

Packet fragmentation can occur for a number of reasons, and, in some cases, it can be controlled through a configuration setting. Every link has a maximum transmission unit (MTU) size that specifies the size of the largest packet that the link can transmit. A larger MTU size means that fewer packets are required to transmit a certain amount of data. However, for a packet to successfully traverse the path from the source node to the destination node, the MTU size of the source node egress interface must be no larger than that of the smallest MTU size of all nodes on the path between the source and destination. This value is referred to as the path maximum transmission unit (path MTU).

When a packet is larger that the MTU size on any link in the data path, the link might fragment it or drop it.

- For IPv4, if a node within the path between a source node and a destination node receives a packet that is larger than its MTU size, it can fragment the packet and transmit the resulting smaller packets.
- For IPv6, an intermediate node cannot fragment a packet. If a packet is larger than a link's MTU size, it is likely that the link will drop it. However, the source node (the node that sent the packet) can fragment a packet, and this is done to accommodate a path MTU size-adjustment requirement. Nodes along the path of a packet cannot fragment the packet to transmit it.

The fragmentation counters feature for IPsec tunnels provides the show output information for the pre-fragments generated and post-fragments generated fields.

Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, SRX5K-MPC3-100G10G (IOC3) and SRX5K-MPC3-40G10G (IOC3) are introduced for SRX5400, SRX5600, and SRX5800 devices that perform hash-based datapath packet forwarding to interconnect with all existing IOC and SPC cards using the XL chip (packet-processing chip). The IOC3 XL chip uses a hash-based method to distribute ingress traffic to a pool of SPUs by default.

Options

- none—Display the security flow statistics information.
- **node**—(Optional) For chassis cluster configurations, display all security flow statistics on a specific node (device) in the cluster.
  - ***node-id***—Identification number of the node. It can be 0 or 1.
  - **all**—Display information about all nodes.
  - **local**—Display information about the local node.
  - **primary**—Display information about the primary node.

Required Privilege Level

view

Related Documentation

- *Understanding Traffic Processing on Security Devices*

List of Sample Output

show security flow statistics on page 1485

Output Fields

Table 182 on page 1484 lists the output fields for the **show security flow statistics** command. Output fields are listed in the approximate order in which they appear.

Table 182: show security flow statistics Output Fields

Field Name	Field Description
Current sessions	Number of active sessions on the SPU.

Table 182: show security flow statistics Output Fields (continued)

Field Name	Field Description
Packets received	Number of packets received in a security flow of a specific SPU. The packets are processed and forwarded on that SPU.
Packets transmitted	Number of packets returned to Jexec for transmission.
Packets forwarded/queued	Number of packets fowarded or number of packets queued up by other modules. NOTE: Dropped packets are not captured by this field.
Packets copied	Number of packets copied by other modules including fragmentation and tcp proxy.
Packets dropped	Number of packets dropped in a flow on a specific SPU. The packets are received in the flow. However, during processing, the system discovers sanity check errors, security violations, or other conditions that caused the packet to be dropped. See the description for some of the conditions and events that can cause a packet to be dropped.
Fragment packets	Number of fragments received in a flow on the SPU. See the description for information about packet fragments.
Pre fragments generated	For IPsec tunnels, the number of fragments that are self-generated by the SRX Series device before it encapsulates the packet with the IPsec encryption header.
Post fragments generated	For IPsec tunnels, the number of fragments that are received by the SRX Series device and packets that are fragmented after encryption.

Sample Output

show security flow statistics

```
user@host> show security flow statistics
node0:
--------------------------------------------------------------------------

    Current sessions: 0
    Packets received: 2677
    Packets transmitted: 2278
    Packets forwarded/queued: 0
    Packets copied: 99
```

```
    Packets dropped: 300
    Fragment packets: 0
    Pre fragments generated: 0
    Post fragments generated: 0

node1:
--------------------------------------------------------------------------

    Current sessions: 0
    Packets received: 1267
    Packets transmitted: 904
    Packets forwarded/queued: 0
    Packets copied: 0
    Packets dropped: 363
    Fragment packets: 0
    Pre fragments generated: 0
    Post fragments generated: 0
```

show security flow status

Syntax **show security flow status**

Release Information Command introduced in Junos OS Release 10.2; session distribution mode option added in Junos OS Release 12.1X44-D10; enhanced route scaling mode option added in Junos OS Release 12.1X45-D10. GTP-U distribution option added in Junos OS Release 15.1X49-D40.
Starting in Junos OS Release 15.1X49-D10 and Junos OS Release 17.3R1, SRX5K-MPC3-100G10G (IOC3) and SRX5K-MPC3-40G10G (IOC3) are introduced for SRX5400, SRX5600, and SRX5800 devices that perform hash-based data path packet forwarding to interconnect with all existing IOC and SPC cards using the XL chip (packet-processing chip).
The IOC3 XL chip uses a hash-based method to distribute ingress traffic to a pool of SPUs by default. Selection of hash keys depends on application protocols.

Description Display the flow processing modes and logging status.

Required Privilege Level view

Related Documentation
- *Understanding Traffic Processing on Security Devices*

List of Sample Output
show security flow status on page 1488
show security flow status (IPsec Performance Acceleration) on page 1488
show security flow status (for hash-based datapath forwarding using SRX5K-MPC3-40G10G (IOC3) and SRX5K-MPC3-100G10G (IOC3) on page 1489

Output Fields Table 183 on page 1487 lists the output fields for the **show security flow status** command. Output fields are listed in the approximate order in which they appear.

Table 183: show security flow status Output Fields

Field Name	Field Description
Flow forwarding mode	Flow processing mode. • Inet forwarding mode • Inet6 forwarding mode • MPLS forwarding mode • ISO forwarding mode • Session distribution mode • Enhanced route scaling mode
Flow trace status	Flow logging status. • Flow tracing status • Flow tracing options

Table 183: show security flow status Output Fields (continued)

Field Name	Field Description
flow session distribution	SPU load distribution mode. • RR-based • Hash-based GTP-U distribution • Enabled
Flow packet ordering	packet-ordering mode. • Hardware • Software
Flow ipsec performance acceleration	IPsec VPN performance acceleration status.

Sample Output

show security flow status

```
root> show security flow status
Flow forwarding mode:
Inet forwarding mode: flow based
Inet6 forwarding mode: flow based
MPLS forwarding mode: drop
ISO forwarding mode: drop
Enhanced route scaling mode: Enabled (reboot needed to disable)
Flow trace status
Flow tracing status: on
Flow tracing options: all
Flow session distribution
Distribution mode: Hash-based
GTP-U distribution: Enabled
Flow packet ordering
Ordering mode: Software (reboot needed to change to software)
```

show security flow status (IPsec Performance Acceleration)

```
root> show security flow status
Flow forwarding mode:
    Inet forwarding mode: flow based
    Inet6 forwarding mode: drop
    MPLS forwarding mode: drop
    ISO forwarding mode: drop
Flow trace status
    Flow tracing status: off
Flow session distribution
    Distribution mode: RR-based
    GTP-U distribution: Enabled Flow packet ordering
Ordering mode: Software (reboot needed to change to software)
Flow ipsec performance acceleration: on
```

show security flow status (for hash-based datapath forwarding using SRX5K-MPC3-40G10G (IOC3) and SRX5K-MPC3-100G10G (IOC3)

```
root> show security flow status
node0:
--------------------------------------------------------------------------
  Flow forwarding mode:
    Inet forwarding mode: flow based
    Inet6 forwarding mode: drop
    MPLS forwarding mode: drop
    ISO forwarding mode: drop
  Flow trace status
    Flow tracing status: off
  Flow session distribution
    Distribution mode: Hash-based
    GTP-U distribution: Enabled
  Flow ipsec performance acceleration: off
  Flow packet ordering
    Ordering mode: Hardware

node1:
--------------------------------------------------------------------------
  Flow forwarding mode:
    Inet forwarding mode: flow based
    Inet6 forwarding mode: drop
    MPLS forwarding mode: drop
    ISO forwarding mode: drop
  Flow trace status
    Flow tracing status: off
  Flow session distribution
    Distribution mode: Hash-based
    GTP-U distribution: Enabled
  Flow ipsec performance acceleration: off
  Flow packet ordering
    Ordering mode: Hardware
```

show security forward-options secure-wire

Syntax show security forward-options secure-wire <*secure-wire-name*>

Release Information Command introduced in Junos OS Release 12.3X48-D10.

Description Display information about secure wire mappings.

Options
- none—Display information about all configured secure wire mappings.
- *secure-wire-name*—(Optional) Display information about the specified secure wire mapping.

Required Privilege Level view

Related Documentation
- Understanding Secure Wire on Security Devices on page 679

List of Sample Output
show security forward-options secure-wire on page 1490
show security forward-options secure-wire pw1 on page 1491

Output Fields Table 184 on page 1490 lists the output fields for the **show security forward-options secure-wire** command. Output fields are listed in the approximate order in which they appear.

Table 184: show security forward-options secure-wire Output Fields

Field Name	Field Description
Secure wire	Name of the secure wire mapping.
Interface	One of the peer interfaces in the secure wire mapping.
Link	Operational status of the interface link.
Interface	The second peer interface in the secure wire mapping.
Link	Operational status of the interface link.

Sample Output

show security forward-options secure-wire

```
user@host> show security forward-options secure-wire
Secure wire                    Interface       Link   Interface       Link

pw1                            ge-11/1/0.0     up     ge-11/1/1.0     up
pw2                            ge-11/0/0.0     up     ge-11/0/1.0     up
```

```
pw3                                ge-11/1/2.0     down    ge-11/1/3.0     down
Total secure wires: 3
```

Sample Output

show security forward-options secure-wire pw1

```
user@host> show security forward-options secure-wire pw1
Secure wire                        Interface       Link    Interface       Link

pw1                                ge-11/1/0.0     up      ge-11/1/1.0     up
```

show security policies

Syntax

show security policies
application-firewall
count
detail
from-zone <*zone-name*>
global
hit-count
interface
logical-system <*logical-system-name*>
policy <*policy-name*>
root-logical-system
service-set
start
tenant <*tenant-name*>
to-zone <*zone-name*>
unknown-source-identity
zone-context

Release Information

Command modified in Junos OS Release 9.2.
Support for IPv6 addresses is added in Junos OS Release 10.2.
Support for wildcard addresses is added in Junos OS Release 11.1.
Support for global policy and services offloading is added in Junos OS Release 11.4.
Support for source-identities and the **Description** output field is added in Junos OS Release 12.1.
Support for negated address added in Junos OS Release 12.1X45-D10.
The output fields for Policy Statistics expanded, and the output fields for the **global** and **policy-name** options are expanded to include from-zone and to-zone global match criteria in Junos OS Release 12.1X47-D10.
Support for the **initial-tcp-mss** and **reverse-tcp-mss** options is added in Junos OS Release 12.3X48-D20.
Output field and description for **source-end-user-profile** option is added in Junos OS Release 15.1x49-D70.
Output field and description for **dynamic-applications** option is added in Junos OS Release 15.1x49-D100.
Output field and description for **dynapp-redir-profile** option is added in Junos OS Release 18.2R1.
The **tenant** option is introduced in Junos OS Release 18.3R1.

Description

Displays a summary of all security policies configured on the device. If a particular policy is specified, display information specific to that policy. The existing show commands for displaying the policies configured with multiple tenant support are enhanced. A security policy controls the traffic flow from one zone to another zone. The security policies allow you to deny, permit, reject (deny and send a TCP RST or ICMP port unreachable message to the source host), encrypt and decrypt, authenticate, prioritize, schedule, filter, and monitor the traffic attempting to cross from one security zone to another.

Options

- **application-firewall**—Displays the information of application-firewall.
- **count**—Displays the number of policies. Range is 1 through 65,535.
- **detail**—(Optional) Displays a detailed view of all of the policies configured on the device.
- **from-zone**—Displays the policy information matching the given source zone.
- **global**—(Optional) Displays information about global policies.
- **hit-count**—Displays the policies hit count.
- **interface**—Displays the name of the adaptive services interface.
- **logical-system**—Displays the logical system name.
- **policy-name**—(Optional) Displays the information about a specified policy.
- **root-logical-system**—Displays root logical system as default.
- **service-set**—Displays the name of the service set.
- **start**—Displays the policies from a given position. Range is 1 through 65,535.
- **tenant**—Displays the name of the tenant system.
- **to-zone**—Displays the policy information matching the given destination zone.
- **unknown-source-identity**—Displays the unknown-source-identity of a policy.
- **zone-context**—Displays the count of policies in each context (from-zone and to-zone).

Required Privilege Level

view

Related Documentation

- *Security Policies Overview*
- *Understanding Security Policy Rules*
- *Understanding Security Policy Elements*
- *Unified Policies Configuration Overview*

List of Sample Output

Output Fields Table 185 on page 1494 lists the output fields for the **show security policies** command. Output fields are listed in the approximate order in which they appear.

Table 185: show security policies Output Fields

Field Name	Field Description
From zone	Name of the source zone.
To zone	Name of the destination zone.
Policy	Name of the applicable policy.
Description	Description of the applicable policy.
State	Status of the policy: • **enabled**: The policy can be used in the policy lookup process, which determines access rights for a packet and the action taken in regard to it. • **disabled**: The policy cannot be used in the policy lookup process, and therefore it is not available for access control.
Index	Internal number associated with the policy.
Sequence number	Number of the policy within a given context. For example, three policies that are applicable in a from-zoneA-to-zoneB context might be ordered with sequence numbers 1, 2, 3. Also, in a from-zoneC-to-zoneD context, four policies might have sequence numbers 1, 2, 3, 4.
Source addresses	For standard display mode, the names of the source addresses for a policy. Address sets are resolved to their individual names. For detail display mode, the names and corresponding IP addresses of the source addresses for a policy. Address sets are resolved to their individual address name-IP address pairs.
Destination addresses	Name of the destination address (or address set) as it was entered in the destination zone's address book. A packet's destination address must match this value for the policy to apply to it.
source-end-user-profile	Name of the device identity profile (referred to as **end-user-profile** in the CLI) that contains attributes, or characteristics of a device. Specification of the device identity profile in the **source-end-user-profile** field is part of the device identity feature. If a device matches the attributes specified in the profile and other security policy parameters, then the security policy's action is applied to traffic issuing from the device.
Source addresses (excluded)	Name of the source address excluded from the policy.
Destination addresses (excluded)	Name of the destination address excluded from the policy.
Source identities	One or more user roles specified for a policy.

Table 185: show security policies Output Fields (continued)

Field Name	Field Description
Applications	Name of a preconfigured or custom application whose type the packet matches, as specified at configuration time. • **IP protocol**: The Internet protocol used by the application—for example, TCP, UDP, ICMP. • **ALG**: If an ALG is explicitly associated with the policy, the name of the ALG is displayed. If **application-protocol ignore** is configured, **ignore** is displayed. Otherwise, **0** is displayed. However, even if this command shows **ALG: 0**, ALGs might be triggered for packets destined to well-known ports on which ALGs are listening, unless ALGs are explicitly disabled or when **application-protocol ignore** is not configured for custom applications. • **Inactivity timeout**: Elapsed time without activity after which the application is terminated. • **Source port range**: The low-high source port range for the session application.
Dynamic Applications	Application identification-based Layer 7 dynamic applications.
Destination Address Translation	Status of the destination address translation traffic: • **drop translated**—Drop the packets with translated destination addresses. • **drop untranslated**—Drop the packets without translated destination addresses.
Application Firewall	An application firewall includes the following: • **Rule-set**—Name of the rule set. • **Rule**—Name of the rule. ◦ **Dynamic applications**—Name of the applications. ◦ **Dynamic application groups**—Name of the application groups. ◦ **Action**—The action taken with respect to a packet that matches the application firewall rule set. Actions include the following: ▪ **permit** ▪ **deny** • **Default rule**—The default rule applied when the identified application is not specified in any rules of the rule set.
Action or Action-type	• The action taken for a packet that matches the policy's tuples. Actions include the following: ◦ **permit** ◦ **firewall-authentication** ◦ **tunnel ipsec-vpn** *vpn-name* ◦ **pair-policy** *pair-policy-name* ◦ **source-nat pool** *pool-name* ◦ **pool-set** *pool-set-name* ◦ **interface** ◦ **destination-nat** *name* ◦ **deny** ◦ **reject** ◦ **services-offload**

Table 185: show security policies Output Fields (continued)

Field Name	Field Description
Session log	Session log entry that indicates whether the **at-create** and **at-close** flags were set at configuration time to log session information.
Scheduler name	Name of a preconfigured scheduler whose schedule determines when the policy is active and can be used as a possible match for traffic.

Policy statistics	• **Input bytes**—The total number of bytes presented for processing by the device.
	  • **Initial direction**—The number of bytes presented for processing by the device from the initial direction.
	  • **Reply direction**—The number of bytes presented for processing by the device from the reply direction.
	• **Output bytes**—The total number of bytes actually processed by the device.
	  • **Initial direction**—The number of bytes from the initial direction actually processed by the device.
	  • **Reply direction**—The number of bytes from the reply direction actually processed by the device.
	• **Input packets**—The total number of packets presented for processing by the device.
	  • **Initial direction**—The number of packets presented for processing by the device from the initial direction.
	  • **Reply direction**—The number of packets presented for processing by the device from the reply direction.
	• **Output packets**—The total number of packets actually processed by the device.
	  • **Initial direction**—The number of packets actually processed by the device from the initial direction.
	  • **Reply direction**—The number of packets actually processed by the device from the reply direction.
	• **Session rate**—The total number of active and deleted sessions.
	• **Active sessions**—The number of sessions currently present because of access control lookups that used this policy.
	• **Session deletions**—The number of sessions deleted since system startup.
	• **Policy lookups**—The number of times the policy was accessed to check for a match.

dynapp-redir-profile	Displays unified policy redirect profile. See *profile(dynamic-application)*.
Per policy TCP Options	Configured syn and sequence checks, and the configured TCP MSS value for the initial direction, the reverse direction or, both.

Sample Output

show security policies

```
user@host> show security policies

  From zone: trust, To zone: untrust
  Policy: p1, State: enabled, Index: 4, Sequence number: 1
    Source addresses:
    sa-1-ipv4: 198.51.100.11/24
    sa-2-ipv6: 2001:db8:a0b:12f0::1/32
```

```
    sa-3-ipv6: 2001:db8:a0b:12f0::22/32
    sa-4-wc:   203.0.113.1/255.255.0.255
    Destination addresses:
    da-1-ipv4: 2.2.2.2/24
    da-2-ipv6: 2001:db8:a0b:12f0::8/32
    da-3-ipv6: 2001:db8:a0b:12f0::9/32
    da-4-wc:   192.168.22.11/255.255.0.255
    Source identities: role1, role2, role4
    Applications: any
    Action: permit, application services, log, scheduled
    Application firewall : my_ruleset1
  Policy: p2, State: enabled, Index: 5, Sequence number: 2
    Source addresses:
    sa-1-ipv4:  198.51.100.11/24
    sa-2-ipv6:  2001:db8:a0b:12f0::1/32
    sa-3-ipv6:  2001:db8:a0b:12f0::22/32
    Destination addresses:
    da-1-ipv4:  2.2.2.2/24
    da-2-ipv6:  2001:db8:a0b:12f0::1/32
    da-3-ipv6:  2001:db8:a0b:12f0::9/32
    Source identities: role1, role4
    Applications: any
    Action: deny, scheduled
```

show security policies (Dynamic Applications)

```
user@host>show security policies

  Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: any
    Dynamic Applications: junos:YAHOO
    Action: deny, log
  Policy: p2, State: enabled, Index: 5, Scope Policy: 0, Sequence number: 2
    Source addresses: any
    Destination addresses: any
    Applications: any
    Dynamic Applications: junos:web, junos:web:social-networking:facebook,
    junos:TFTP, junos:QQ
    Action: permit, log
  Policy: p3, State: enabled, Index: 6, Scope Policy: 0, Sequence number: 3
    Source addresses: any
    Destination addresses: any
    Applications: any
    Dynamic Applications: junos:HTTP, junos:SSL
    Action: permit, application services, log
```

The following example displays the output with unified policies configured.

```
user@host> show security policies

Default policy: deny-all
Pre ID default policy: permit-all
From zone: trust, To zone: untrust
  Policy: p2, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: junos-defaults
    Dynamic Applications: junos:GMAIL, junos:FACEBOOK-CHAT
    dynapp-redir-profile: profile1
```

show security policies policy-name detail

```
user@host> show security policies policy-name p1 detail

  Policy: p1, action-type: permit, State: enabled, Index: 4, Scope Policy: 0
  Description: The policy p1 is for the sales team
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses:
    sa-1-ipv4:  198.51.100.11/24
    sa-2-ipv6:  2001:db8:a0b:12f0::1/32
    sa-3-ipv6:  2001:db8:a0b:12f0::9/32
    sa-4-wc:    203.0.113.1/255.255.0.255
  Destination addresses:
    da-1-ipv4: 192.0.2.0/24
    da-2-ipv6: 2001:db8:a0b:12f0::1/32
    da-3-ipv6: 2001:db8:a0b:12f0::9/32
    da-4-wc:   192.168.22.11/255.255.0.255
 Source identities:
  role1
  role2
  role4
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Destination Address Translation: drop translated
  Application firewall :
  Rule-set: my_ruleset1
    Rule: rule1
      Dynamic Applications: junos:FACEBOOK-ACCESS, junos:YMSG
      Dynamic Application groups: junos:web, junos:chat
      Action: deny
    Default rule: permit
  Session log: at-create, at-close
  Scheduler name: sch20
  Per policy TCP Options: SYN check: No, SEQ check: No
  Policy statistics:
    Input  bytes       :               18144                 545 bps
      Initial direction:                9072                272 bps
      Reply direction  :                9072                272 bps
    Output bytes       :               18144                 545 bps
      Initial direction:                9072                272 bps
      Reply direction  :                9072                272 bps
    Input  packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Output packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Session rate       :                 108                   3 sps
    Active sessions    :                  93
    Session deletions  :                  15
    Policy lookups     :                 108
```

The following example displays the output with unified policies configured.

```
user@host> show security policies policy-name p1 detail

Default policy: permit-all
Pre ID default policy: permit-all
```

```
From zone: trust, To zone: trust
  Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses: any
    Destination addresses: any
    Applications: any
    Action: reject
    dynapp-redir-profile: profile1
```

show security policies (Services-Offload)

```
user@host> show security policies

  Policy: p1, action-type: reject, State: enabled, Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: trust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  dynapp-redir-profile: profile1(1)
  Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
```

show security policies (Device Identity)

```
user@host> show security policies
From zone: trust, To zone: untrust
       Policy: dev-id-marketing, State: enabled, Index: 5, Scope Policy: 0,
Sequence number: 1
         Source addresses: any
         Destination addresses: any
         source-end-user-profile: marketing-profile
         Applications: any
         Action: permit
```

show security policies detail

```
user@host> show security policies detail

Default policy: deny-all
Policy: p1, action-type: permit, services-offload:enabled , State: enabled, Index:
 4, Scope Policy: 0
  Policy Type: Configured
  Description: The policy p1 is for the sales team
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
 Source identities:
  role1
```

```
  role2
  role4
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Per policy TCP Options: SYN check: No, SEQ check: No
  Policy statistics:
    Input  bytes       :               18144                 545 bps
      Initial direction:                9072                272 bps
      Reply direction  :                9072                272 bps
    Output bytes       :               18144                 545 bps
      Initial direction:                9072                272 bps
      Reply direction  :                9072                272 bps
    Input  packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Output packets     :                 216                   6 pps
      Initial direction:                 108                   3 bps
      Reply direction  :                 108                   3 bps
    Session rate       :                 108                   3 sps
    Active sessions    :                  93
    Session deletions  :                  15
    Policy lookups     :                 108
Policy: p2, action-type: permit, services-offload:enabled , State: enabled, Index:
 5, Scope Policy: 0
  Policy Type: Configured
  Description: The policy p2 is for the sales team
  Sequence number: 1
  From zone: untrust, To zone: trust
  Source addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global):  0.0.0.0/0
    any-ipv6(global): ::/0
 Source identities:
  role1
  role2
  role4
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Per policy TCP Options: SYN check: No, SEQ check: No
```

The following example displays the output with unified policies configured.

```
user@host> show security policies detail

Default policy: deny-all
Pre ID default policy: permit-all
Policy: p2, action-type: reject, State: enabled, Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
```

```
    any-ipv6(global): ::/0
  Application: junos-defaults
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [443-443]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [5432-5432]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [80-80]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [3128-3128]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [8000-8000]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [8080-8080]
    IP protocol: 17, ALG: 0, Inactivity timeout: 60
      Source port range: [0-0]
      Destination port range: [1-65535]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [443-443]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [5432-5432]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [80-80]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [3128-3128]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [8000-8000]
    IP protocol: 6, ALG: 0, Inactivity timeout: 1800
      Source port range: [0-0]
      Destination port range: [8080-8080]
    IP protocol: 17, ALG: 0, Inactivity timeout: 60
      Source port range: [0-0]
      Destination port range: [1-65535]
  Dynamic Application:
    junos:FACEBOOK-CHAT: 10704
    junos:GMAIL: 51
  dynapp-redir-profile: profile1(1)
  Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
```

show security policies detail (TCP Options)

```
user@host> show security policies policy-name p2 detail
node0:
--------------------------------------------------------------------------
Policy:p2, action-type:permit, State: enabled,Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: trust
  Source addresses:
    any-ipv4(global): 0.0.0.0/0
```

```
    any-ipv6(global): ::/0
  Destination addresses:
    any-ipv4(global): 0.0.0.0/0
    any-ipv6(global): ::/0
  Application: junos-defaults
    IP protocol: tcp, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
 Destination port range: [80-80]
Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
  Dynamic-application: junos:HTTP
```

show security policies policy-name (Negated Address)

```
user@host> show security policies policy-name p1
node0:
--------------------------------------------------------------------------
From zone: trust, To zone: untrust
  Policy: p1, State: enabled, Index: 4, Scope Policy: 0, Sequence number: 1
    Source addresses(excluded): as1
    Destination addresses(excluded): as2
    Applications: any
    Action: permit
```

show security policies policy-name detail (Negated Address)

```
user@host> show security policies policy-name p1 detail
node0:
--------------------------------------------------------------------------
Policy: p1, action-type: permit, State: enabled, Index: 4, Scope Policy: 0
  Policy Type: Configured
  Sequence number: 1
  From zone: trust, To zone: untrust
  Source addresses(excluded):
    ad1(ad): 255.255.255.255/32
    ad2(ad):  198.51.100.1/24
    ad3(ad):  198.51.100.6  ~ 198.51.100.56
    ad4(ad):  192.0.2.8/24
    ad5(ad):  198.51.100.99 ~ 198.51.100.199
    ad6(ad):  203.0.113.9/24
    ad7(ad):  203.0.113.23/24
  Destination addresses(excluded):
    ad13(ad2): 198.51.100.76/24
    ad12(ad2): 198.51.100.88/24
    ad11(ad2): 192.0.2.23 ~ 192.0.2.66
    ad10(ad2): 192.0.2.93
    ad9(ad2):  203.0.113.76 ~ 203.0.113.106
    ad8(ad2):  203.0.113.199
  Application: any
    IP protocol: 0, ALG: 0, Inactivity timeout: 0
      Source port range: [0-0]
      Destination port range: [0-0]
  Per policy TCP Options: SYN check: No, SEQ check: No
```

show security policies global

```
user@host> show security policies global policy-name Pa
node0:
--------------------------------------------------------------------------
  Global policies:
  Policy: Pa, State: enabled, Index: 6, Scope Policy: 0, Sequence number: 1
```

```
From zones: any
To zones: any
Source addresses: H0
Destination addresses: H1
Applications: junos-http
Action: permit
```

show security policies detail tenant

```
user@host> show security policies detail tenant TN1

Default policy: deny-all
Pre ID default policy: permit-all
Policy: p1, action-type: permit, State: enabled, Index: 4, Scope Policy: 0
Policy Type: Configured
Sequence number: 1
From zone: trust, To zone: untrust
Source addresses: any
Destination addresses: any
Application: junos-ping
IP protocol: 1, ALG: 0, Inactivity timeout: 60
ICMP Information: type=255, code=0
Application: junos-telnet
IP protocol: tcp, ALG: 0, Inactivity timeout: 1800
Source port range: [0-0]
Destination port range: [23-23]
Application: app_udp
IP protocol: udp, ALG: 0, Inactivity timeout: 1800
Source port range: [0-0]
Destination port range: [5000-5000]
Application: junos-icmp6-all
IP protocol: 58, ALG: 0, Inactivity timeout: 60
ICMP Information: type=255, code=0
Per policy TCP Options: SYN check: No, SEQ check: No, Window scale: No
Session log: at-create, at-close
Policy statistics:
Input  bytes     :                   0                   0 bps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Output bytes     :                   0                   0 bps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Input  packets   :                   0                   0 pps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Output packets   :                   0                   0 pps
Initial direction:                   0                   0 bps
Reply direction  :                   0                   0 bps
Session rate     :                   0                   0 sps
Active sessions  :                   0
Session deletions:                   0
Policy lookups   :                   0
```

show security zones

Syntax

show security zones <*zone-name*>
detail
logical-system <*logical-system-name*>
root-logical-system
tenant <*tenant-name*>
terse
type

Release Information

Command introduced in Junos OS Release 8.5.
The **Description** output field added in Junos OS Release 12.1.
The **tenant** option is introduced in Junos OS Release 18.3R1.

Description

Displays the information about the security zones. You can define a security zone, which allows you to divide the network into different segments and apply different security options to each segment. The existing show commands for displaying the zones configured with multiple tenant support are enhanced.

Options

- **detail**—(Optional) Displays the detail level of output.
- **terse**—(Optional) Displays the specified level of output.
- ***zone-name***—(Optional) Displays information about the specified zone.
- **logical-system**—Displays logical system name.
- **root-logical-system**—Displays root logical system as default.
- **tenant**—Displays the name of the tenant system.
- **type**—Displays the information for zones of a specified type.

Required Privilege Level

view

Related Documentation

- *Security Zones Overview*
- *Supported System Services for Host Inbound Traffic*
- security-zone on page 1101

List of Sample Output

show security zones on page 1505
show security zones abc on page 1506
show security zones abc detail on page 1506
show security zones terse on page 1506
show security zone tenant all on page 1506

Output Fields

Table 186 on page 1505 lists the output fields for the **show security zones** command. Output fields are listed in the approximate order in which they appear.

Table 186: show security zones Output Fields

Field Name	Field Description	Level of Output
Functional zone	Name of the functional zone.	none
Security zone	Name of the security zone.	detail none
Description	Description of the security zone.	detail none
Policy configurable	Whether the policy can be configured or not.	detail none
Interfaces bound	Number of interfaces in the zone.	detail none
Interfaces	List of the interfaces in the zone.	detail none
Zone	Name of the zone.	terse
Type	Type of the zone.	terse
Tenant	Name of the tenant system.	detail

Sample Output

show security zones

```
user@host> show security zones
Functional zone: management
   Description: This is the management zone.
   Policy configurable: No
   Interfaces bound: 1
   Interfaces:
     ge-0/0/0.0
Security zone: Host
   Description: This is the host zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     fxp0.0
Security zone: abc
   Description: This is the abc zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
```

```
    ge-0/0/1.0
Security zone: def
   Description: This is the def zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     ge-0/0/2.0
```

Sample Output

show security zones abc

```
user@host> show security zones abc
Security zone: abc
   Description: This is the abc zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     ge-0/0/1.0
```

Sample Output

show security zones abc detail

```
user@host> show security zones abc detail
Security zone: abc
   Description: This is the abc zone.
   Send reset for non-SYN session TCP packets: Off
   Policy configurable: Yes
   Interfaces bound: 1
   Interfaces:
     ge-0/0/1.0
```

Sample Output

show security zones terse

```
user@host> show security zones terse
Zone                    Type
my-internal             Security
my-external             Security
dmz                     Security
```

show security zone tenant all

```
user@host  show security zone tenant all

  Tenant: TN1


  Security zone: Host
  Send reset for non-SYN session TCP packets: Off
  Policy configurable: Yes
  Interfaces bound: 0
  Interfaces:

  Security zone: abc
```

```
Send reset for non-SYN session TCP packets: Off
Policy configurable: Yes
Interfaces bound: 0
Interfaces:xe-0/0/1.0

Security zone: def
Send reset for non-SYN session TCP packets: Off
Policy configurable: Yes
Interfaces bound: 1
Interfaces:xe-0/0/3.0
```

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 on page 629
- Verifying That Proxy ARP Is Working Correctly on page 634

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

List of Syntax

Syntax (EX Series and QFX Series Switches) on page 1510
Syntax (EX Series with ELS Switches and MX Routers) on page 1510
Syntax (SRX Devices) on page 1510

Syntax (EX Series and QFX Series Switches)

```
show vlans
<brief | detail | extensive>
<dot1q-tunneling>
<management-vlan>
<sort-by (tag | name)>
<vlan-range-name>
<summary>
<vlan-name>
<vlan-range-name>
```

Syntax (EX Series with ELS Switches and MX Routers)

```
show vlans
<brief | detail | extensive>
<instance instance-name>
<logical-system logical-system-name>
<operational>
<vlan-name)>
<interface interface-name>
```

Syntax (SRX Devices)

```
show vlans
<brief | detail | extensive>
<interface interface-name>
<logical-system (logical-system | all)>
<operational>
```

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.
Option **dot1q-tunneling** added in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 12.3R2 for EX Series switches.
Option **interface** introduced in Junos OS Release 13.2X50-D10 (ELS).

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 For EX Series and QFX Series switches:

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.

dot1q-tunneling—(Optional) Display VLANs with the Q-in-Q tunneling feature enabled.

management-vlan—(Optional) Display management VLANs.

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.

summary—(Optional) Display the total number of VLANs and counts of VLANs by type—for example, the number of dynamic, 802.1Q-tagged, and Q-in-Q tunneled VLANs.

vlan-range-name—(Optional) Display information for the specified VLAN range. To display information for all members of the VLAN range, specify the base VLAN name—for example, **employee** for a VLAN range that includes **__employee_1__** through **__employee_10__**.

For EX Series with ELS Switches and MX Routers:

none—Display information for all VLANs.

brief | detail | extensive—(Optional) Display the specified level of output.

instance ***instance-name***—(Optional) Display information for the specified routing instance.

logical-system ***logical-system-name***—(Optional) Display Ethernet-switching statistics information for the specified logical system.

operational—(Optional) Display information for the operational routing instances.

***vlan-name*)**— (Optional) Display information about the specified VLAN.

interface ***interface-name***—(Optional) Display information about the specified interface.

For SRX devices:

none—Display information for all VLANs.

brief | detail | extensive—(Optional) Display the specified level of output.

interface *interface-name*— (Optional) Display information about a specific interface.

logical system—(Optional) Display name of the logical system or all.

operational—(Optional) Display information for the operational switching instances.

Required Privilege Level

view

Related Documentation

- Example: Setting Up Basic Bridging and a VLAN on Switches on page 104
- Example: Setting Up Bridging with Multiple VLANs on page 141
- Example: Setting Up Basic Bridging and a VLAN for an EX Series Switch on page 122
- Example: Setting Up Bridging with Multiple VLANs for EX Series Switches on page 153
- Example: Configuring a Private VLAN on a Single EX Series Switch on page 284
- Example: Configuring a Private VLAN Spanning Multiple EX Series Switches on page 326
- Example: Setting Up Q-in-Q Tunneling on EX Series Switches on page 601
- Understanding Bridging and VLANs on Switches on page 84
- show ethernet-switching interfaces on page 1215

List of Sample Output

show vlans (SRX devices) on page 1528
show vlans brief (SRX devices) on page 1529
show vlans detail (SRX devices) on page 1529

Output Fields Table 187 on page 1513 lists the output fields for the **show vlans** command. Output fields are listed in the approximate order in which they appear.

Table 187: 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**
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**

Table 187: show vlans Output Fields (continued)

Field Name	Field Description	Level of Output
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 187: 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 (EX Series and QFX Series Switches)

show vlans (EX Series and QFX Series)

```
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 on EX and QFX Series)

```
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 (EX and QFX Series)

```
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 (EX Series and QFX Series)

```
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 (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__
```

show vlans extensive (Port-Based on EX Series and QFX Series)

```
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 extensive (MAC-based)

```
user@switch> show vlans extensive
VLAN: default, Created at: Thu May 15 13:43:09 2008
Internal index: 3, Admin State: Enabled, Origin: Static
Protocol: Port Mode, Mac aging time: 300 seconds
Number of interfaces: Tagged 0 (Active = 0), Untagged  2 (Active = 2)
      ge-0/0/0.0*, untagged, access
      ge-0/0/14.0*, untagged, access

VLAN: vlan_dyn, Created at: Thu May 15 13:43:09 2008
Internal index: 4, Admin State: Enabled, Origin: Static
Protocol: Port Mode
Number of interfaces: Tagged 0 (Active = 0), Untagged  0 (Active = 0)
Protocol: MAC Based
```

```
Number of MAC entries: 6
      ge-0/0/0.0*
                  00:00:00:00:00:02 (untagged)
                  00:00:00:00:00:03 (untagged)
                  00:00:00:00:00:04 (untagged)
                  00:00:00:00:00:05 (untagged)
                  00:00:00:00:00:06 (untagged)
                  00:00:00:00:00:07 (untagged)
```

show vlans (Q-in-Q Tunneling on EX Series and QFX Series)

```
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 on EX Series and QFX Series)

```
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 on EX Series and QFX Series)

```
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 (EX Series and QFX Series)

```
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 (EX Series and QFX Series)

```
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 (EX Series and QFX Series)

```
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*
```

show vlans sort-by tag (EX Series)

```
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 employee (vlan-range-name)

```
user@switch> show vlans employee

Name              Tag   Interfaces

__employee_120__ 120
                            ge-0/0/22.0*
__employee_121__ 121
                            ge-0/0/22.0*
__employee_122__ 122
                            ge-0/0/22.0*
__employee_123__ 123
                            ge-0/0/22.0*
__employee_124__ 124
                            ge-0/0/22.0*
__employee_125__ 125
                            ge-0/0/22.0*
__employee_126__ 126
                            ge-0/0/22.0*
__employee_127__ 127
                            ge-0/0/22.0*
__employee_128__ 128
                            ge-0/0/22.0*
__employee_129__ 129
                            ge-0/0/22.0*
__employee_130__ 130
                            ge-0/0/22.0*
```

show vlans summary (EX Series)

```
user@switch> show vlans summary
VLANs summary:
         Total: 8,   Configured VLANs: 5
         Internal VLANs: 1,  Temporary VLANs: 0

     Dot1q VLANs summary:
         Total: 8,  Tagged VLANs: 2, Untagged VLANs: 6
         Private VLAN:
             Primary VLANs: 2,  Community VLANs: 2, Isolated VLANs: 3

     Dot1q Tunneled VLANs summary:
         Total: 0
         Private VLAN:
             Primary VLANs: 0,  Community VLANs: 0, Isolated VLANs: 0

     Dynamic VLANs:
        Total: 2,  Dot1x: 2, MVRP: 0
```

Sample Output: EX Series with ELS Switches and MX Routers

show vlans brief (EX Series Switch)

```
user@switch> show vlans brief
Routing instance        VLAN name             Tag          Interfaces
default-switch          c1                    20
                                                           ge-0/0/0.0*
                                                           ge-1/0/0.0*
                                                           ge-2/0/0.0*
default-switch          c2                    30
                                                           ge-0/0/0.0*
                                                           ge-2/0/0.0*
default-switch          default               1

default-switch          iso                   10
                                                           ge-0/0/1.0*
default-switch          iso1                  50
                                                           ge-0/0/0.0*
                                                           ge-2/0/0.0*
default-switch          pri                   100
                                                           ge-0/0/0.0*
                                                           ge-1/0/0.0*
                                                           ge-2/0/0.0*
```

show vlans brief (MX Routers)

```
user@host> show vlans brief
Routing instance        VLAN name             Tag         Interfaces
VPLS-1                  __VPLS-1__               all
                                                            ae1.0
VPLS-2                  __VPLS-2__               all
                                                            ae3.0
                                                            ge-3/1/2.0
                                                            vt-3/3/10.1048576
default-switch          VLAN1000                 1000
                                                            ae26.0
default-switch          VLAN101                  101
                                                            ae20.0
default-switch          VLAN102                  102
                                                            ae20.0
default-switch          VLAN103                  103
                                                            ae20.0
default-switch          VLAN104                  104
                                                            ae20.0
default-switch          VLAN105                  105
                                                            ae20.0
default-switch          VLAN106                  106
                                                            ae20.0
default-switch          VLAN107                  107
                                                            ae20.0
default-switch          VLAN108                  108
                                                            ae20.0
[...output truncated...]
```

show vlans detail (EX Series Switch)

```
user@switch> show vlans detail
```

```
Routing instance: default-switch
  VLAN Name: c1                               State: Active
Tag: 20
PVLAN type : Community
Internal index: 16, Generation Index: 21, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: c2                               State: Active
Tag: 30
PVLAN type : Community
Internal index: 17, Generation Index: 22, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: default                          State: Active
Tag: 1
Internal index: 5, Generation Index: 5, Origin: Static
MAC aging time: 300 seconds
Number of interfaces: Tagged 0    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: iso                              State: Active
Tag: 10
Internal index: 14, Generation Index: 19, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/1.0*,untagged,access
Number of interfaces: Tagged 0    , Untagged 1
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: iso1                             State: Active
Tag: 50
PVLAN type : Isolated
Internal index: 15, Generation Index: 20, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: pri                              State: Active
Tag: 100
PVLAN type : Primary
Isolated VLAN :
vlan-id : 50 vlan name : iso1
```

```
Community VLAN :
vlan-id : 20 vlan name : c1
vlan-id : 30 vlan name : c2
Internal index: 9, Generation Index: 14, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0
```

show vlans detail (MX Routers)

```
user@host> show vlans detail
Routing instance: VPLS-1
  VLAN Name: __VPLS-1__                     State: Active
Tag: all
Internal index: 2, Generation Index:  , Origin: Dynamic
Interfaces:
    ae1.0,tagged
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0

Routing instance: VPLS-2
  VLAN Name: __VPLS-2__                     State: Active
Tag: all
Internal index: 3, Generation Index:  , Origin: Dynamic
Interfaces:
    ae3.0,tagged
    ge-3/1/2.0,tagged
    vt-3/3/10.1048576,tagged
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 4

Routing instance: default-switch
  VLAN Name: VLAN1000                       State: Active
Tag: 1000
Internal index: 4, Generation Index: 1, Origin: Static
Layer 3 interface: irb.1000
Interfaces:
    ae26.0,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: VLAN101                        State: Active
Tag: 101
Internal index: 5, Generation Index: 2, Origin: Static
Layer 3 interface: irb.101
Interfaces:
    ae20.0,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 1

Routing instance: default-switch
  VLAN Name: VLAN102                        State: Active
Tag: 102
Internal index: 6, Generation Index: 3, Origin: Static
Layer 3 interface: irb.102
```

```
Interfaces:
    ae20.0,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 1
[...output truncated...]
```

show vlans extensive (EX Series Switch)

```
user@switch> show vlans extensive
Routing instance: default-switch
  VLAN Name: c1                             State: Active
Tag: 20
PVLAN type : Community
Internal index: 16, Generation Index: 21, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: c2                             State: Active
Tag: 30
PVLAN type : Community
Internal index: 17, Generation Index: 22, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: default                        State: Active
Tag: 1
Internal index: 5, Generation Index: 5, Origin: Static
MAC aging time: 300 seconds
Number of interfaces: Tagged 0    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: iso                            State: Active
Tag: 10
Internal index: 14, Generation Index: 19, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/1.0*,untagged,access
Number of interfaces: Tagged 0    , Untagged 1
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: iso1                           State: Active
Tag: 50
PVLAN type : Isolated
Internal index: 15, Generation Index: 20, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
```

```
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: pri                            State: Active
Tag: 100
PVLAN type : Primary
Isolated VLAN :
vlan-id : 50 vlan name : iso1
Community VLAN :
vlan-id : 20 vlan name : c1
vlan-id : 30 vlan name : c2
Internal index: 9, Generation Index: 14, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-0/0/0.0*,tagged,trunk
    ge-1/0/0.0*,tagged,trunk
    ge-2/0/0.0*,tagged,trunk
Number of interfaces: Tagged 3    , Untagged 0
Total MAC count: 0
```

show vlans extensive (MX Routers)

```
user@host> show vlans extensive
Routing instance: default-switch
  VLAN Name: VLAN_10                        State: Active
Tag: 10
Internal index: 2, Generation Index: 1, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-1/0/3.0*,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: VLAN_20                        State: Active
Tag: 20
Internal index: 3, Generation Index: 2, Origin: Static
MAC aging time: 300 seconds
Interfaces:
    ge-1/0/3.0*,tagged,trunk
Number of interfaces: Tagged 1    , Untagged 0
Total MAC count: 0
```

Sample Output (SRX Devices)

show vlans (SRX devices)

```
user@host> show vlans
Routing instance        VLAN name             Tag          Interfaces
default-switch          vlan-22                 22
                                                           ge-0/0/1.0*
default-switch          vlan-333                333
                                                           ge-0/0/3.0*
                                                           ge-0/0/4.0*
default-switch          default               1
                                                           ge-0/0/2.0*
default-switch          vlan100               100
```

```
                                                                ge-0/0/1.0*
```

show vlans brief (SRX devices)

```
user@host> show vlans brief
Routing instance        VLAN name             Tag          Interfaces
default-switch          vlan-22                 22
                                                           ge-0/0/3.0*
                                                           ge-0/0/4.0*
default-switch          default               1
                                                           ge-0/0/1.0*
```

show vlans detail (SRX devices)

```
user@host> show vlans detail
Routing instance: default-switch
  VLAN Name: vlan-22                            State: Active
Tag: 22
Internal index: 2, Generation Index: 1, Origin: Static
MAC aging time: 300 seconds
VXLAN Enabled : No
Number of interfaces: Tagged 0    , Untagged 0
Total MAC count: 0

Routing instance: default-switch
  VLAN Name: vlan-333                           State: Active
Tag: 333
Internal index: 3, Generation Index: 2, Origin: Static
MAC aging time: 300 seconds
VXLAN Enabled : No
Interfaces:
    ge-0/0/3.0*,tagged,trunk
    ge-0/0/4.0*,tagged,trunk
Number of interfaces: Tagged 2    , Untagged 0
Total MAC count: 0
```

traceroute ethernet

Syntax

traceroute ethernet
local-mep *mep-id*
maintenance-association *ma-name*
maintenance-domain *md-name*
<**ttl** *value*>
<**wait** *seconds*>
mac-address | *mep-id*
<**detail**>

Release Information

Command introduced in Junos OS Release 9.0.
mep-id option introduced in Junos OS Release 9.1.
local-mep option introduced in Junos OS Release 15.1

Description

Triggers the linktrace protocol to trace the route between two maintenance points. The result of the traceroute protocol is stored in the path database. To display the path database, use the **show oam ethernet connectivity-fault-management path-database** command.

Before using the traceroute command, you can verify the remote MEP's MAC address using the **show oam ethernet connectivity-fault-management path-database** command.

Options

local-mep ***mep-id***—(Required when multiple MEPs are configured) Identifier for the local maintenance endpoint.

detail—(Optional) Provide detailed information of the responder hostname, ingress port name, egress port name, TTL, and relay action.

mac-address—Destination unicast MAC address of the remote maintenance point.

mep-id—MEP identifier of the remote maintenance point. The range of values is 1 through 8191.

maintenance-association ***ma-name***—Specifies an existing maintenance association from the set of configured maintenance associations.

maintenance-domain ***md-name***—Specifies an existing maintenance domain from the set of configured maintenance domains.

ttl ***value***—Number of hops to use in the linktrace request. The range is 1 to 255 hops. The default is 4.

wait ***seconds***—(Optional) Maximum time to wait for a response to the traceroute request. The range is 1 to 255 seconds. The default is 5.

Required Privilege Level

network

List of Sample Output

traceroute ethernet on page 1532

traceroute ethernet detail on page 1532

Output Fields Table 188 on page 1531 lists the output fields for the **traceroute ethernet** command. Output fields are listed in the approximate order in which they appear.

Table 188: traceroute ethernet Output Fields

Field Name	Field Description
Linktrace to	MAC address of the destination maintenance point.
Interface	Local interface used to send the linktrace message (LTM).
Maintenance Domain	Maintenance domain specified in the traceroute command.
Level	Maintenance domain level configured.
Maintenance Association	Maintenance association specified in the traceroute command.
Local Mep	The local maintenance end point identifier.
Transaction Identifier	4-byte identifier maintained by the MEP. Each LTM uses a transaction identifier. The transaction identifier is maintained globally across all Maintenance Domains. Use the transaction identifier to match an incoming linktrace response (LTR), with a previously sent LTM.
Hop	Sequential hop count of the linktrace path.
TTL	Number of hops remaining in the linktrace message. The time to live (TTL) is decremented at each hop.
Source MAC address	MAC address of the 802.1ag node responding to the LTM or the source MAC address of the LTR.
Next-hop MAC address	MAC address of the egress interface of the node to which the LTM is forwarded or the next-hop MAC address derived from the next egress identifier in the Egress-ID TLV of the LTR PDU.
Responder Hostname	The hostname of the responding router. A valid hostname is received only when the responding system is a Juniper Networks router.
Ingress port name	The port name for ingress connections.
Egress port name	The port name for egress connections.

Table 188: traceroute ethernet Output Fields (continued)

Field Name	Field Description
Flags	The configurable flags can include: • H— Hardware only, incoming LT frame has hardware bit set. • T— Terminal MEP, responder is a terminating MEP. • F— FWD yes, LTM frame is relayed further.
Relay Action	The associated relay action. Relay action can be one of the following: • RlyHit— Relay hit; target MAC address matches the MP mac address. • RlyFDB— Relay FDB; output port decided by consulting forwarding database. • RlyMPDB— Relay MIP; output port decided by consulting MIP database.

Sample Output

traceroute ethernet

```
user@host> traceroute ethernet maintenance-domain md1 maintenance-association ma1
00:01:02:03:04:05
 Linktrace to 00:01:02:03:04:05, Interface : ge-5/0/0.0
       Maintenance Domain: MD1, Level: 7
       Maintenance Association: MA1, Local Mep: 1

       Hop     TTL      Source MAC address        Next hop MAC address
       Transaction Identifier:100001
       1       63       00:00:aa:aa:aa:aa         00:00:ab:ab:ab:ab
       2       62       00:00:bb:bb:bb:bb         00:00:bc:bc:bc:bc
       3       61       00:00:cc:cc:cc:cc         00:00:cd:cd:cd:cd
       4       60       00:01:02:03:04:05         00:00:00:00:00:00
```

traceroute ethernet detail

```
user@host> run traceroute ethernet maintenance-domain md6 maintenance-association ma6
mep 101 detail
Linktrace to 00:00:5E:00:53:CC, Interface : ge-1/0/0.1
Maintenance Domain: md6, Level: 6
Maintenance Association: ma6, Local Mep: 201
Transaction Identifier: 2077547465

Legend for RelayAction:
RlyHit  -- Relay hit, Target MAC address matches the MP mac address
RlyFDB  -- Relay FDB, output port decided by consulting FDB database
RlyMPDB -- Relay MIP, output port decided by consulting MIP database

Legend for Flags:
H -- Hardware only,incoming LT frame has hardware bit set
T -- Terminal MEP, responder is a terminating MEP
F -- FWD yes, LTM frame is relayed further

TTL   Responder Hostname    Ingress port name     Egress port name
RelayAction
```

```
Responder Service     Ingress MAC address   Egress MAC address     Flags
62    host1                 ge-1/0/0.1            ge-2/3/0.1             RlyFDB
br1                   00:00:5E:00:53:00     00:00:5E:00:53:A0      HF-
63    host2                 ge-2/3/0.1            ge-1/0/0.1             RlyFDB
br1                   00:00:5E:00:53:AA     00:00:5E:00:53:A2      HF-
61    host3                 ge-1/0/0.1            --:--                  RlyHit
br1                   00:00:5E:00:53:B0           --:--            H-T
```