Table of Contents

About the Documentation ....................................................... xi
Documentation and Release Notes ........................................ xi
Using the Examples in This Manual ....................................... xi
  Merging a Full Example ................................................. xii
  Merging a Snippet ........................................................... xii
Documentation Conventions ................................................. xiii
Documentation Feedback ..................................................... xv
  Requesting Technical Support ........................................... xv
  Self-Help Online Tools and Resources ................................ xvi
  Opening a Case with JTAC ................................................ xvi

Chapter 1 Understanding Layer 2 Bridging, Address Learning, and Forwarding .............................................. 17
  Understanding Layer 2 Bridge Domains .................................. 17
  Understanding Layer 2 Learning and Forwarding ....................... 18
  Layer 2 Bridge Domains on ACX Series Overview ....................... 19
  Layer 2 Learning and Forwarding for Bridge Domains Overview ........ 22

Chapter 2 Configuring Layer 2 Bridging and Layer 3 IP Routing ................................................................. 25
  Configuring a Bridge Domain ............................................ 25
  Configuring a Bridge Domain on ACX Series Routers ................... 28
  Example: Configuring Basic Layer 2 Switching on MX Series ............ 29
  Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances ............................................... 38
  Configuring VLAN Identifiers for Bridge Domains in ACX Series .......... 44
  Configuring Bridge Domains as Switches for Layer 2 Trunk Ports ........ 44

Chapter 3 Configuring Layer 2 Virtual Switches .................................................. 47
  Understanding Layer 2 Virtual Switches .................................. 47
  Configuring a Layer 2 Virtual Switch .................................... 48
  Configuring a Virtual Switch Routing Instance on MX Series Routers ....... 49
  Configuring Integrated Routing and Bridging for a Bridge Domain in a Layer 2 Virtual Switch ........................................ 50
  Configuring Integrated Routing and Bridging in ACX Series ............... 51
  Configuring VPLS Ports in a Virtual Switch ................................ 55
  Configuring a Layer 2 Virtual Switch with a Layer 2 Trunk Port .......... 57

Chapter 4 Configuring Layer 2 Address Learning and Forwarding ................................................................. 61
  Configuring the MAC Table Timeout Interval ............................ 61
  Enabling MAC Accounting .................................................. 62
  Limiting the Number of MAC Addresses Learned from Each Logical Interface .......... 63
  Disabling Layer 2 Learning and Forwarding ............................. 64
  Example: Loop Detection Using the MAC Move Approach ................ 64
List of Figures

<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Configuring Layer 2 Bridging and Layer 3 IP Routing</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Figure 1: Basic Layer 2 Switching</td>
<td>30</td>
</tr>
</tbody>
</table>
# List of Tables

About the Documentation ........................................... xi

Table 1: Notice Icons .............................................. xiii
Table 2: Text and Syntax Conventions .......................... xiv

**Chapter 1** Understanding Layer 2 Bridging, Address Learning, and Forwarding ........... 17

Table 3: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain ........................................... 21
Table 4: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain ........................................... 21

**Chapter 2** Configuring Layer 2 Bridging and Layer 3 IP Routing ......................... 25

Table 5: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain ........................................... 42
Table 6: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain ........................................... 43

**Chapter 11** Operational Mode Commands for Layer 2 Bridge Domains ................. 157

Table 7: show bridge mac-table Output Fields .......................................... 171
Table 8: Layer 2 Overhead and Transmitted Packets or Byte Counts ...................... 180
Table 9: show interfaces queue Output Fields .......................................... 183
Table 10: Byte Count by Interface Hardware .......................................... 187

**Chapter 12** Operational Mode Commands for Layer 2 Learning ......................... 223

Table 11: show l2-learning global-information Output Fields .......................... 225
Table 12: show l2-learning instance Output Fields ..................................... 228
Table 13: show l2-learning interface Output Fields ..................................... 230
About the Documentation

- Documentation and Release Notes on page xi
- Using the Examples in This Manual on page xi
- Documentation Conventions on page xiii
- Documentation Feedback on page xv
- Requesting Technical Support on page xv

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;
       }
     }
   }
   ```

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 xiii* defines notice icons used in this guide.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon-info.png" alt="Info" /></td>
<td>Informational note</td>
<td>Indicates important features or instructions.</td>
</tr>
<tr>
<td><img src="icon-caution.png" alt="Caution" /></td>
<td>Caution</td>
<td>Indicates a situation that might result in loss of data or hardware damage.</td>
</tr>
<tr>
<td><img src="icon-warning.png" alt="Warning" /></td>
<td>Warning</td>
<td>Alerts you to the risk of personal injury or death.</td>
</tr>
<tr>
<td><img src="icon-laser.png" alt="Laser" /></td>
<td>Laser warning</td>
<td>Alerts you to the risk of personal injury from a laser.</td>
</tr>
<tr>
<td><img src="icon-tip.png" alt="Tip" /></td>
<td>Tip</td>
<td>Indicates helpful information.</td>
</tr>
<tr>
<td><img src="icon-best-practice.png" alt="Best practice" /></td>
<td>Best practice</td>
<td>Alerts you to a recommended use or implementation.</td>
</tr>
</tbody>
</table>

*Table 2 on page xiv* defines the text and syntax conventions used in this guide.
### Table 2: Text and Syntax Conventions

<table>
<thead>
<tr>
<th><strong>Convention</strong></th>
<th><strong>Description</strong></th>
<th><strong>Examples</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold text like this</strong></td>
<td>Represents text that you type.</td>
<td>To enter configuration mode, type the <code>configure</code> command:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>user@host&gt; configure</code></td>
</tr>
<tr>
<td><strong>Fixed-width text like this</strong></td>
<td>Represents output that appears on the terminal screen.</td>
<td><code>user@host&gt; show chassis alarms</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No alarms currently active</td>
</tr>
<tr>
<td><strong>Italic text like this</strong></td>
<td>• Introduces or emphasizes important new terms.</td>
<td>• A policy term is a named structure that defines match conditions and actions.</td>
</tr>
<tr>
<td></td>
<td>• Identifies guide names.</td>
<td>• <code>Junos OS CLI User Guide</code></td>
</tr>
<tr>
<td></td>
<td>• Identifies RFC and internet draft titles.</td>
<td>• RFC 1997, <code>BGP Communities Attribute</code></td>
</tr>
<tr>
<td><strong>Italic text like this</strong></td>
<td>Represents variables (options for which you substitute a value) in commands or configuration statements.</td>
<td>Configure the machine’s domain name:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>[edit] root@# set system domain-name domain-name</code></td>
</tr>
<tr>
<td><strong>Text like this</strong></td>
<td>Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.</td>
<td>• To configure a stub area, include the <code>stub</code> statement at the <code>[edit protocols ospf area area-id]</code> hierarchy level.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The console port is labeled <code>CONSOLE</code>.</td>
</tr>
<tr>
<td>&lt; &gt; (angle brackets)</td>
<td>Encloses optional keywords or variables.</td>
<td><code>stub &lt;default-metric metric&gt;;</code></td>
</tr>
<tr>
<td></td>
<td>(pipe symbol)</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>(pound sign)</td>
<td>Indicates a comment specified on the same line as the configuration statement to which it applies.</td>
</tr>
<tr>
<td>] (square brackets)</td>
<td>Encloses a variable for which you can substitute one or more values.</td>
<td><code>community name members [ community-ids ]</code></td>
</tr>
<tr>
<td>Indentation and braces ( { } )</td>
<td>Identifies a level in the configuration hierarchy.</td>
<td><code>[edit] routing-options { </code> rout; return; <code> }</code></td>
</tr>
<tr>
<td>: (semicolon)</td>
<td>Identifies a leaf statement at a configuration hierarchy level.</td>
<td></td>
</tr>
</tbody>
</table>
Table 2: Text and Syntax Conventions (continued)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI Conventions</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bold text like this</strong></td>
<td>Represents graphical user interface (GUI) items you click or select.</td>
<td>• In the Logical Interfaces box, select <strong>All Interfaces</strong>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• To cancel the configuration, click <strong>Cancel</strong>.</td>
</tr>
<tr>
<td><strong>&gt; (bold right angle bracket)</strong></td>
<td>Separates levels in a hierarchy of menu selections.</td>
<td>In the configuration editor hierarchy, select <strong>Protocols &gt; Ospf</strong>.</td>
</tr>
</tbody>
</table>

**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.

- 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.
CHAPTER 1

Understanding Layer 2 Bridging, Address Learning, and Forwarding

- Understanding Layer 2 Bridge Domains on page 17
- Understanding Layer 2 Learning and Forwarding on page 18
- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22

Understanding Layer 2 Bridge Domains

You can configure one or more bridge domains on MX Series routers to perform Layer 2 bridging. The Layer 2 bridging functions of the MX Series routers 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 bridge domain is a set of logical ports that share the same flooding or broadcast characteristics. Like a virtual LAN (VLAN), a bridge domain spans one or more ports of multiple devices.

On Juniper Networks MX Series 5G Universal Routing Platforms only, you can configure one or more bridge domains to perform Layer 2 bridging. Thus, MX Series routers 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 bridge domain. 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 bridge domain that has a Layer 3 protocol configured.

You can also group one or more bridge domains within a single instance, or virtual switch. The MX Series routers also support multiple virtual switches, each of which operates independently of other virtual switches on the router. Virtual switches isolate a LAN segment with its spanning-tree protocol instance. Thus, each virtual switch can participate in a different Layer 2 network.

In Junos OS Release 9.2 and later, bridge domains 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 bridge
domains 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 bridge domain that has the same VLAN identifier. A Layer 2 trunk interface also supports IRB within a bridge domain. In addition, you can configure Layer 2 learning and forwarding properties that apply to the entire set of bridge domains.

In Junos OS Release 9.3 and later, 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 bridge domains in the virtual switch can handle VPLS routing instance traffic. Packets received on a Layer 2 trunk interface are forwarded within a bridge domain that has the same VLAN identifier.

**Understanding Layer 2 Learning and Forwarding**

On MX Series routers only, you can configure Layer 2 MAC address and VLAN learning and forwarding properties in support of Layer 2 bridging. The router learns unicast media access control (MAC) addresses to avoid flooding the packets to all the ports in a bridge domain. The MX Series router creates a source MAC entry in its source and destination MAC tables for each MAC address learned from packets received on ports that belong to the bridge domain. If the bridge domain receives a control protocol data unit (PDU) which does not have a corresponding protocol configured, then the control PDU is considered as an unknown multicast data packet and the packets are flooded across all the ports that are part of the same bridge domain. If the bridge domain has the protocol corresponding to the PDU configured, then the control PDU is considered as a control packet and is processed by the routing engine.

By default, Layer 2 address learning is enabled. You can disable MAC learning for the router or for a specific bridge domain or logical interfaces. You can also configure the following Layer 2 forwarding properties for an MX Series router:

- Timeout interval for MAC entries
- MAC accounting
- A limit to the number of MAC addresses learned from the logical interfaces

**Related Documentation**

- Understanding Layer 2 Virtual Switches on page 47
- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Configuring a Bridge Domain on page 25

- Understanding Layer 2 Bridge Domains on page 17
- Configuring the MAC Table Timeout Interval on page 61
- Enabling MAC Accounting on page 62
- Limiting the Number of MAC Addresses Learned from Each Logical Interface on page 63

Copyright © 2018, Juniper Networks, Inc.
Layer 2 Bridge Domains on ACX Series Overview

A bridge domain is a set of logical interfaces that share the same flooding or broadcast characteristics. Layer 2 logical interfaces are created by defining one or more logical units on a physical interface with encapsulation as ethernet-bridge or vlan-bridge. All the member ports of the bridge domain participate in Layer 2 learning and forwarding. You can configure one or more bridge domains on ACX Series routers to perform Layer 2 bridging. The Layer 2 bridging functions of ACX Series routers include integrated routing and bridging (IRB) 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 bridge domain that has a Layer 3 protocol configured.

NOTE: ACX Series routers do not support the creation of bridge domains by using access and trunk ports.

You can configure E-LAN and E-LINE services by using bridge domains.

On ACX Series routers, you can configure bridge domains by using the following methods:

- Bridge domain without a `vlan-id number` statement
- Bridge domain with the `vlan-id` value set to `none`
- Bridge domain with a single `vlan-id`
- Bridge domain with a `vlan-id-list`

NOTE: The Layer 2 CLI configurations and show commands for ACX5048 and ACX5096 routers differ compared to other ACX Series routers. For more information, see Layer 2 Next Generation Mode for ACX Series.

When you configure E-LAN and E-LINE services using a bridge domain without a `vlan-id number` statement, the bridge domain should explicitly be normalized to a service VLAN ID and TPID by configuring an input VLAN map under a logical interface. Explicit normalization is required when a logical interface’s outer VLAN ID and TPID is not the same as the service VLAN ID and TPID of the service being configured using a bridge domain.

The following input VLAN map functions are supported in ACX Series routers:

- `push`—Add a new VLAN tag to the top of the VLAN stack.
- `swap`—Replace the outer VLAN tag of the VLAN stack in a frame.
- `pop`—Remove a VLAN tag from the top of the VLAN tag stack.
• **swap-swap**—Replace both the outer and inner VLAN tags of the frame.

• **push-push**—Push two VLAN tags on top of the VLAN stack.

**NOTE:** push-push does not work on ACX Series routers if the incoming packet already has a VLAN tag.

The following VLAN map functions are not supported in ACX Series routers:

• **swap-push**—Replace the outer VLAN tag of the frame and add a new VLAN tag to the top of the VLAN stack.

• **pop-swap**—Remove the outer VLAN tag of the frame and replace the inner VLAN tag of the frame.

• **pop-pop**—Remove both the outer and inner VLAN tags of the frame.

**NOTE:** You can configure Q-in-Q tunneling by explicitly configuring an input VLAN map with the push function on the ingress logical interface.

A bridge domain can also be created by using aggregated Ethernet interfaces. Aggregated Ethernet interfaces are considered as logical interfaces in a bridge domain.

The following steps outline the process for bridging a packet received over a Layer 2 logical interface:

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. If the bridge domain is configured without a `vlan-id number` statement, then the VLAN tags are rewritten based on the input VLAN map configured on the logical interface and normalized to a service VLAN ID.

3. If the bridge domain is configured with a normalizing VLAN identifier by using the `vlan-id number` statement, the VLAN tags of the received packet are 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 3 on page 21.

4. 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.

5. 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.
6. If the bridge domain is configured without a `vlan-id number` statement, then for each outbound Layer 2 logical interface, the VLAN tags are rewritten based on the output VLAN map configured on that logical interface.

7. If the bridge domain is configured with a normalizing VLAN identifier by using the `vlan-id number` statement, for each outbound Layer 2 logical interface, the normalizing VLAN identifier configured for the bridge domain 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 bridge domain, the VLAN tags are rewritten as described in Table 4 on page 21.

Table 3 on page 21 shows specific examples of how the VLAN tags of packets sent to the bridge domain 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 3: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain

<table>
<thead>
<tr>
<th>VLAN Identifier of Logical Interface</th>
<th>VLAN Configurations for Bridge Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>vlan-id none</td>
<td>vlan-id 200</td>
</tr>
<tr>
<td>none</td>
<td>No operation</td>
</tr>
<tr>
<td>200</td>
<td>pop 200</td>
</tr>
<tr>
<td>1000</td>
<td>pop 1000</td>
</tr>
<tr>
<td>vlan-tags outer 2000, inner 300</td>
<td>pop 2000, pop 300</td>
</tr>
<tr>
<td>vlan-tags outer 100, inner 400</td>
<td>pop 100, pop 400</td>
</tr>
<tr>
<td>vlan-id-range 10-100</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 4 on page 21 shows specific examples of how the VLAN tags for packets sent from the bridge domain 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 4: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain

<table>
<thead>
<tr>
<th>VLAN Identifier of Logical Interface</th>
<th>VLAN Configurations for Bridge Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>vlan-id none</td>
<td>vlan-id 200</td>
</tr>
<tr>
<td>none</td>
<td>no operation</td>
</tr>
<tr>
<td>200</td>
<td>pop 200</td>
</tr>
</tbody>
</table>
### Table 4: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain (continued)

<table>
<thead>
<tr>
<th>VLAN Identifier of Logical Interface</th>
<th>VLAN Configurations for Bridge Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>vlan-id none</td>
<td>vlan-id 200</td>
</tr>
<tr>
<td>200</td>
<td>push 200</td>
</tr>
<tr>
<td></td>
<td>No operation</td>
</tr>
<tr>
<td>1000</td>
<td>push 1000</td>
</tr>
<tr>
<td></td>
<td>swap 200 to 1000</td>
</tr>
<tr>
<td>vlan-tags outer 2000 inner 300</td>
<td>push 2000, push 300</td>
</tr>
<tr>
<td></td>
<td>swap 200 to 300, push 2000</td>
</tr>
<tr>
<td>vlan-tags outer 100 inner 400</td>
<td>push 100, push 400</td>
</tr>
<tr>
<td></td>
<td>swap 200 to 400, push 100</td>
</tr>
<tr>
<td>vlan-id-range 10-100</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>--</td>
</tr>
</tbody>
</table>

#### Limitations on Layer 2 bridging

The following Layer 2 bridging limitations apply for ACX Series Universal Metro Routers:

- A bridge domain cannot have two or more logical interfaces that belong to the same physical interface.
- A bridge domain with dual VLAN ID tag is not supported.
- The maximum number of supported input VLAN maps with TPID `swap` is 64.
- MAC learning cannot be disabled at a logical interface level.
- MAC limit per logical interface cannot be configured.

#### Related Documentation

- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72

#### Layer 2 Learning and Forwarding for Bridge Domains Overview

When you configure a bridge domain, Layer 2 address learning is enabled by default. The bridge domain learns unicast media access control (MAC) addresses to avoid flooding...
the packets to all the ports in the bridge domain. Each bridge domain 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 bridge domain.

**NOTE:** Traffic is not flooded back onto the interface on which it was received.

You can optionally disable MAC learning either for the entire router or for a specific bridge domain. You can also configure the following Layer 2 learning and forwarding properties:

- Static MAC entries on logical interfaces
- Size of the MAC address table for the bridge domain

**Related Documentation**

- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72
Configuring Layer 2 Bridging and Layer 3 IP Routing

• Configuring a Bridge Domain on page 25
• Configuring a Bridge Domain on ACX Series Routers on page 28
• Example: Configuring Basic Layer 2 Switching on MX Series on page 29
• Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances on page 38
• Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
• Configuring Bridge Domains as Switches for Layer 2 Trunk Ports on page 44

Configuring a Bridge Domain

A bridge domain must include a set of logical interfaces that participate in Layer 2 learning and forwarding. You can optionally configure a VLAN identifier and a routing interface for the bridge domain to also support Layer 3 IP routing.

To enable a bridge domain, include the following statements:

```
[edit]
bridge-domains {
    bridge-domain-name {
        domain-type bridge:
        interface interface-name;
        routing-interface routing-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 bridge domain 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. For information about VLAN identifiers and VLAN tags for a bridge domain, see “Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances” on page 38.
To include one or more logical interfaces in the bridge domain, specify an `interface-name` for an Ethernet interface you configured at the `[edit interfaces]` hierarchy level.

**NOTE:** A maximum of 4000 active logical interfaces are supported on a bridge domain or on each mesh group in a virtual private LAN service (VPLS) instance configured for Layer 2 bridging.

By default, each bridge domain maintains a Layer 2 forwarding database that contains media access control (MAC) addresses learned from packets received on the ports that belong to the bridge domain. You can modify Layer 2 forwarding properties, including disabling MAC learning for the entire system or a bridge domain, adding static MAC addresses for specific logical interfaces, and limiting the number of MAC addresses learned by the entire system, the bridge domain, or a logical interface.

You can also configure spanning tree protocols to prevent forwarding loops.

In Junos OS Release 8.5 and later, you can configure IGMP snooping for a bridge domain. For more information, see the *Multicast Protocols Feature Guide*.

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 bridge domain that has an IRB interface configured. You configure a logical routing interface by including the `irb` statement at the `[edit interfaces]` hierarchy level and include that interface in the bridge domain. For more information about how to configure a routing interface, see the Junos OS Network Interfaces Library for Routing Devices.

**NOTE:** You can include only one routing interface in a bridge domain.

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

```plaintext
[edit]
bridge-domains {
  bridge-domain-name {
    domain-type bridge;
    interface interface-name;
    routing-interface routing-interface-name;
    service-id number;
    vlan-id (none | number);
    vlan-tags outer number inner number;
  }
}
```

For each bridge domain that you configure, specify a `bridge-domain-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 routing interface to support IRB in a bridge domain, 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 bridge domain, you can include either the vlan-id statement or the vlan-tags statement, but not both.

For MC-LAG bridge domains, when the VLAN identifier is none, use the service-id statement to facilitate media access control (MAC) and Address Resolution Protocol (ARP) synchronization among MC-LAG peers.

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

NOTE: A maximum of 4000 active logical interfaces are supported on a bridge domain or on each mesh group in a VPLS routing instance configured for Layer 2 bridging.

To associate a routing interface with a bridge domain, include the routing-interface routing-interface-name statement and specify a routing-interface-name you configured at the [edit interfaces irb] hierarchy level. You can configure only one routing interface for each bridge domain. For more information about how to configure logical and routing interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

In Junos OS Release 9.0 and later, IRB interfaces are supported for multicast snooping. For more information about multicast snooping, see the Understanding Multicast Snooping and VPLS Root Protection.

In Junos 11.4 and later, IP multicast is supported on Layer 2 trunk ports through IRB interfaces using the Trio chipset.

In Junos OS Release 9.6 and later, 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 two options, ce and irb. 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. For more information about configuring VPNs, see the Junos VPN Configuration Guide.
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.

Integrated Bridging and Routing (IRB) interfaces are used to tie together Layer 2 switched and Layer 3 routed domains on MX routers. MX routers support classifiers and rewrite rules on the IRB interface at the [edit class-of-service interfaces irb unit logical-unit-number] level of the hierarchy. All types of classifiers and rewrite rules are allowed, including IEEE 802.1p.

NOTE: The IRB classifiers and rewrite rules are used only for routed packets; in other words, it is for traffic that originated in the Layer 2 domain and is then routed through IRB into the Layer 3 domain, or vice versa. Only IEEE classifiers and IEEE rewrite rules are allowed for pure Layer 2 interfaces within a bridge domain.

Related Documentation
• Understanding Layer 2 Learning and Forwarding on page 18
• Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
• Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83

Configuring a Bridge Domain on ACX Series Routers

A bridge domain must include a set of logical interfaces that participate in Layer 2 learning and forwarding.

To configure a bridge domain, include the following statements:

```
[edit]
bridge-domains {
  bridge-domain-name {
    interface interface-name;
    vlan-id (none | number);  
    vlan-id-list [ vlan-id-numbers ];
  }
}
```

You cannot use the slash (/) character in bridge domain 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 none.

To include one or more logical interfaces in the bridge domain, specify an interface name for an Ethernet interface you configured at the [edit bridge-domains bridge-domain-name] hierarchy level.
To configure a layer 2 logical interface to be included in a bridge domain, you can either include the `encapsulation vlan-bridge` statement under the logical interface, or the `encapsulation ethernet-bridge` statement under the physical interface.

**NOTE:** A maximum of 1000 logical interfaces can be configured on a physical interface. You can configure a maximum of 3000 bridge domains on an ACX Series router.

---

**Related Documentation**
- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72

---

**Example: Configuring Basic Layer 2 Switching on MX Series**

This example shows how to configure Layer 2 switching with all interfaces participating in a single VLAN.

- **Requirements** on page 29
- **Overview** on page 29
- **Configuration** on page 30
- **Verification** on page 32

**Requirements**

No special configuration beyond device initialization is required before configuring this example.

This example uses an MX Series device to perform Layer 2 switching.

**Overview**

In this example, a single MX Series device is configured to act as a basic single-VLAN switch. Three connections are in place. The connections from the MX Series device attach to Junos OS routers, but the routers are used here for testing purposes only. In place of routers, you can use any IP networking devices.
Topology

Figure 1 on page 30 shows the sample network.

Figure 1: Basic Layer 2 Switching

"CLI Quick Configuration" on page 30 shows the configuration for all of the devices in Figure 1 on page 30.

The section “Step-by-Step Procedure” on page 31 describes the steps on Device S1.

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, and then copy and paste the commands into the CLI at the `[edit]` hierarchy level.

**Device S1**

```
set interfaces ge-2/0/0 vlan-tagging
set interfaces ge-2/0/0 encapsulation extended-vlan-bridge
set interfaces ge-2/0/0 unit 0 vlan-id 600
set interfaces ge-2/0/1 vlan-tagging
set interfaces ge-2/0/1 encapsulation extended-vlan-bridge
set interfaces ge-2/0/1 unit 0 vlan-id 600
set interfaces ge-2/0/2 vlan-tagging
set interfaces ge-2/0/2 encapsulation extended-vlan-bridge
set interfaces ge-2/0/2 unit 0 vlan-id 600
set bridge-domains customer1 domain-type bridge
set bridge-domains customer1 interface ge-2/0/0.0
set bridge-domains customer1 interface ge-2/0/2.0
set bridge-domains customer1 interface ge-2/0/1.0
```

**Device R1**

```
set interfaces ge-1/3/2 vlan-tagging
set interfaces ge-1/3/2 unit 0 vlan-id 600
set interfaces ge-1/3/2 unit 0 family inet address 10.0.0.1/24
```

**Device R2**

```
set interfaces ge-3/1/0 vlan-tagging
set interfaces ge-3/1/0 unit 0 vlan-id 600
set interfaces ge-3/1/0 unit 0 family inet address 10.0.0.2/24
```
Device R3

```
set interfaces ge-2/0/1 vlan-tagging
set interfaces ge-2/0/1 unit 0 vlan-id 600
set interfaces ge-2/0/1 unit 0 family inet address 10.0.0.3/24
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure Device S1:

1. Configure the device interfaces.

   ```
   [edit interfaces]
   user@S1# set interfaces ge-2/0/0 vlan-tagging
   user@S1# set interfaces ge-2/0/0 encapsulation extended-vlan-bridge
   user@S1# set interfaces ge-2/0/0 unit 0 vlan-id 600
   user@S1# set interfaces ge-2/0/1 vlan-tagging
   user@S1# set interfaces ge-2/0/1 encapsulation extended-vlan-bridge
   user@S1# set interfaces ge-2/0/1 unit 0 vlan-id 600
   user@S1# set interfaces ge-2/0/2 vlan-tagging
   user@S1# set interfaces ge-2/0/2 encapsulation extended-vlan-bridge
   user@S1# set interfaces ge-2/0/2 unit 0 vlan-id 600
   ```

2. Configure the bridge domain.

   ```
   [edit interfaces]
   user@S1# set bridge-domains customer1 domain-type bridge
   user@S1# set bridge-domains customer1 interface ge-2/0/0.0
   user@S1# set bridge-domains customer1 interface ge-2/0/2.0
   user@S1# set bridge-domains customer1 interface ge-2/0/1.0
   ```

Results

From configuration mode, confirm your configuration by entering the `show interfaces` and `show bridge-domains` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@S1# show interfaces
ge-2/0/0 {
  vlan-tagging;
  encapsulation extended-vlan-bridge;
  unit 0 {
    vlan-id 600;
  }
}
ge-2/0/1 {
  vlan-tagging;
  encapsulation extended-vlan-bridge;
  unit 0 {
    vlan-id 600;
  }
}
```
ge-2/0/2 {
    vlan-tagging;
    encapsulation extended-vlan-bridge;
    unit 0 {
        vlan-id 600;
    }
}

user@S1# show bridge-domains
customer1 {
    domain-type bridge;
    interface ge-2/0/0.0;
    interface ge-2/0/2.0;
    interface ge-2/0/1.0;
}
If you are done configuring the device, enter commit from configuration mode.

Verification

Confirm that the configuration is working properly.

- Confirming the MAC Address Learning on page 32
- Making Sure That the Attached Devices Can Reach Each Other on page 33
- Checking the Bridge Domain on page 34
- Checking the Bridge Statistics on page 35
- Checking the Bridge Flooding on page 36
- Checking Layer 2 Learning on page 37

Confirming the MAC Address Learning

Purpose Display Layer 2 MAC address information.

Action • From Device S1, run the show bridge mac-table command.

user@S1> show bridge mac-table

| MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC) |
|-----------------|-----------------|-----------------|
| MAC address     | MAC flags       | Logical interface |
| 00:12:1e:ee:34:dd | D               | ge-2/0/2.0      |
| 00:1d:b5:5e:86:79 | D               | ge-2/0/0.0      |
| 00:21:59:0f:35:2b | D               | ge-2/0/1.0      |

• From Device S1, run the show bridge mac-table extensive command.
user@S1> show bridge mac-table extensive

MAC address: 00:12:1e:ee:34:dd
  Routing instance: default-switch
  Bridging domain: customer1, VLAN : NA
  Learning interface: ge-2/0/2.0
  Layer 2 flags: in_hash, in_ifd, in_ifl, in_vlan, in_rtt, kernel, in_ifbd
  Epoch: 1                            Sequence number: 0
  Learning mask: 0x00000004

MAC address: 00:1d:b5:5e:86:79
  Routing instance: default-switch
  Bridging domain: customer1, VLAN : NA
  Learning interface: ge-2/0/0.0
  Layer 2 flags: in_hash, in_ifd, in_ifl, in_vlan, in_rtt, kernel, in_ifbd
  Epoch: 1                            Sequence number: 0
  Learning mask: 0x00000004

MAC address: 00:21:59:0f:35:2b
  Routing instance: default-switch
  Briding domain: customer1, VLAN : NA
  Learning interface: ge-2/0/1.0
  Layer 2 flags: in_hash, in_ifd, in_ifl, in_vlan, in_rtt, kernel, in_ifbd
  Epoch: 3                            Sequence number: 0
  Learning mask: 0x00000004

Meaning  The output shows that the MAC addresses have been learned.

Making Sure That the Attached Devices Can Reach Each Other

Purpose  Verify connectivity.
<table>
<thead>
<tr>
<th>Action</th>
<th>user@R1&gt; ping 10.0.0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PING 10.0.0.2 (10.0.0.2): 56 data bytes</td>
</tr>
<tr>
<td></td>
<td>64 bytes from 10.0.0.2: icmp_seq=0 ttl=64 time=1.178 ms</td>
</tr>
<tr>
<td></td>
<td>64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1.192 ms</td>
</tr>
<tr>
<td></td>
<td>64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=1.149 ms</td>
</tr>
<tr>
<td></td>
<td>^C</td>
</tr>
<tr>
<td></td>
<td>--- 10.0.0.2 ping statistics ---</td>
</tr>
<tr>
<td></td>
<td>3 packets transmitted, 3 packets received, 0% packet loss</td>
</tr>
<tr>
<td></td>
<td>round-trip min/avg/max/stddev = 1.149/1.173/1.192/0.018 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>user@R1&gt; ping 10.0.0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PING 10.0.0.3 (10.0.0.3): 56 data bytes</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=0 ttl=64 time=1.189 ms</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=1.175 ms</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=1.178 ms</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=1.133 ms</td>
</tr>
<tr>
<td>^C</td>
</tr>
<tr>
<td>--- 10.0.0.3 ping statistics ---</td>
</tr>
<tr>
<td>4 packets transmitted, 4 packets received, 0% packet loss</td>
</tr>
<tr>
<td>round-trip min/avg/max/stddev = 1.133/1.169/1.189/0.021 ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>user@R2&gt; ping 10.0.0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PING 10.0.0.3 (10.0.0.3): 56 data bytes</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=0 ttl=64 time=0.762 ms</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.651 ms</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.722 ms</td>
</tr>
<tr>
<td>64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.705 ms</td>
</tr>
<tr>
<td>^C</td>
</tr>
<tr>
<td>--- 10.0.0.3 ping statistics ---</td>
</tr>
<tr>
<td>4 packets transmitted, 4 packets received, 0% packet loss</td>
</tr>
<tr>
<td>round-trip min/avg/max/stddev = 0.651/0.710/0.762/0.040 ms</td>
</tr>
</tbody>
</table>

**Meaning**

The output shows that the attached devices have established Layer 3 connectivity, with Device S1 doing transparent Layer 2 bridging.

**Checking the Bridge Domain**

**Purpose**

Display bridge domain information.
Action  
```bash
user@S1> show bridge domain extensive
```

```
Routing instance: default-switch
Bridge domain: customer1                      State: Active
Bridge VLAN ID: NA
Interfaces:
  ge-2/0/0.0
  ge-2/0/1.0
  ge-2/0/2.0
Total MAC count: 3
```

Meaning  
The output shows that bridge domain is active.

**Checking the Bridge Statistics**

Purpose  
Display bridge statistics.
Action  
user@S1>  show bridge statistics

<table>
<thead>
<tr>
<th>Interface</th>
<th>Broadcast packets</th>
<th>Broadcast bytes</th>
<th>Multicast packets</th>
<th>Multicast bytes</th>
<th>Flooded packets</th>
<th>Flooded bytes</th>
<th>Unicast packets</th>
<th>Unicast bytes</th>
<th>Current MAC count</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-2/0/0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>64</td>
<td>1 (Limit 1024)</td>
</tr>
<tr>
<td>ge-2/0/2.0</td>
<td>0</td>
<td>0</td>
<td>80</td>
<td>8160</td>
<td>0</td>
<td>0</td>
<td>52</td>
<td>4332</td>
<td>1 (Limit 1024)</td>
</tr>
<tr>
<td>ge-2/0/1.0</td>
<td>2</td>
<td>128</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>93</td>
<td>51</td>
<td>4249</td>
<td>1 (Limit 1024)</td>
</tr>
</tbody>
</table>

Meaning  
The output shows that bridge domain interfaces are sending and receiving packets.

Checking the Bridge Flooding

Purpose  
Display bridge flooding information.
Action  

`user@S1> show bridge flood extensive`

Name: __juniper_private1__
CEs: 0
VEs: 0
Name: default-switch
CEs: 3
VEs: 0
Bridging domain: customer1
Flood route prefix: 0x30003/51
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: 568
Flooding to:
- Name: __all_ces__ Type: Group, NhType: comp, Index: 562
  Composition: split-horizon
Children:
- Name: ge-2/0/0.0 Type: CE, NhType: ucst, Index: 524
- Name: ge-2/0/1.0 Type: CE, NhType: ucst, Index: 513
- Name: ge-2/0/2.0 Type: CE, NhType: ucst, Index: 523

Meaning

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 bridge domain 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.

Checking Layer 2 Learning

Purpose

Display Layer 2 learning information for all the interfaces.
Action

user@S1> show l2-learning interface

Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop, LH - MAC limit hit, DN - Interface Down )

<table>
<thead>
<tr>
<th>Logical Interface Name</th>
<th>BD</th>
<th>MAC</th>
<th>STP</th>
<th>Logical Interface flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-2/0/2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>custom..</td>
<td>1024</td>
<td></td>
<td></td>
<td>Forwarding</td>
</tr>
</tbody>
</table>

Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop, LH - MAC limit hit, DN - Interface Down )

<table>
<thead>
<tr>
<th>Logical Interface Name</th>
<th>BD</th>
<th>MAC</th>
<th>STP</th>
<th>Logical Interface flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-2/0/0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>custom..</td>
<td>1024</td>
<td></td>
<td></td>
<td>Forwarding</td>
</tr>
</tbody>
</table>

Routing Instance Name : default-switch
Logical Interface flags (DL - disable learning, AD - packet action drop, LH - MAC limit hit, DN - Interface Down )

<table>
<thead>
<tr>
<th>Logical Interface Name</th>
<th>BD</th>
<th>MAC</th>
<th>STP</th>
<th>Logical Interface flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-2/0/1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>custom..</td>
<td>1024</td>
<td></td>
<td></td>
<td>Forwarding</td>
</tr>
</tbody>
</table>

Related Documentation

- Understanding OSPF Areas
- Examples: Configuring OSPF Stub and Not-So-Stubby Areas

Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances

For a bridge domain that is performing Layer 2 switching only, you do not have to specify a VLAN identifier.

For a bridge domain 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.

You can configure VLAN identifiers for a bridge domain or a VPLS routing instance in the following ways:

- By using the `input-vlan-map` and the `output-vlan-map` statements at the `[edit interfaces interface-name]` or `[edit logical-systems logical-system-name interfaces interface-name]` hierarchy level to configure VLAN mapping. For information about configuring input and output VLAN maps to stack and rewrite VLAN tags in incoming or outgoing frames, see the Junos OS Network Interfaces Library for Routing Devices.

- 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 bridge domain or a VPLS routing instance.
The `vlan-id` and `vlan-tags` statements are used to specify the normalizing VLAN identifier under the bridge domain or VPLS routing instance. The normalizing VLAN identifier is used to perform the following functions:

- Translate, or normalize, the VLAN tags of received 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 bridge domain or VPLS routing instance using the `vlan-id` or `vlan-tags` statements.

To configure a VLAN identifier for a bridge domain, include either the `vlan-id` or the `vlan-tags` statement at the `[edit interfaces interface-name unit logic-unit-number family bridge]` or `[edit logical-systems logical-system-name interfaces interface-name unit logic-unit-number family bridge]` hierarchy level, and then include that logical interface in the bridge domain configuration. For more information about configuring a bridge domain, see “Configuring a Bridge Domain” on page 25.

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. For more information about configuring a VPLS routing instance, see the Junos OS VPNs Library for Routing Devices.

**NOTE:** The maximum number of Layer 2 interfaces that you can associate with a bridge domain or a VPLS instance on MX Series routers is 4000.

**NOTE:** For a single bridge domain 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 bridge domain 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 3 on page 21. 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 bridge domain 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 bridge domain within a routing instance. Each bridge domain 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 4 on page 21.

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:

```plaintext
[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-instance {
   V1 {
      instance-type vpls;
      vlan-id all;
      interface ge-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 doubly-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 bridge domain 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 3 on page 21.

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 bridge domain 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 bridge domain or VPLS routing instance, the VLAN tags are rewritten as described in Table 4 on page 21.

The tables below show how VLAN tags are applied for traffic sent to and from the bridge domain, depending on how the `vlan-id` and `vlan-tags` statements are configured for the bridge domain and on how VLAN identifiers are configured for the logical interfaces in a bridge domain 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 3 on page 21 shows specific examples of how the VLAN tags for packets sent to the bridge domain 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 5: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain

<table>
<thead>
<tr>
<th>VLAN Identifier of Logical Interface</th>
<th>VLAN Configurations for Bridge Domain</th>
<th>vlan tags outer 100</th>
<th>inner 300</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN Configurations</td>
<td></td>
<td>vlan-id none</td>
<td>vlan-id 200</td>
</tr>
<tr>
<td>none</td>
<td>No operation</td>
<td>push 200</td>
<td>–</td>
</tr>
<tr>
<td>200</td>
<td>pop 200</td>
<td>No operation</td>
<td>No operation</td>
</tr>
<tr>
<td>1000</td>
<td>pop 1000</td>
<td>swap 1000 to 200</td>
<td>No operation</td>
</tr>
<tr>
<td>vlan-tags outer 2000</td>
<td></td>
<td>pop 2000, pop 300</td>
<td>pop 2000, swap 300 to 200</td>
</tr>
<tr>
<td>inner 300</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5: Statement Usage and Input Rewrite Operations for VLAN Identifiers for a Bridge Domain (continued)

<table>
<thead>
<tr>
<th>VLAN Identifier of Logical Interface</th>
<th>VLAN Configurations for Bridge Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vlan-id none</td>
</tr>
<tr>
<td>vlan-tags outer 100 inner 400</td>
<td>pop 100, pop 400</td>
</tr>
<tr>
<td>vlan-id-range 10-100</td>
<td>–</td>
</tr>
<tr>
<td>vlan-tags outer 200 inner-range 10-100</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 4 on page 21 shows specific examples of how the VLAN tags for packets sent from the bridge domain 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 6: Statement Usage and Output Rewrite Operations for VLAN Identifiers for a Bridge Domain

<table>
<thead>
<tr>
<th>VLAN Identifier of Logical Interface</th>
<th>VLAN Configurations for Bridge Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>vlan-id none</td>
</tr>
<tr>
<td>none</td>
<td>no operation</td>
</tr>
<tr>
<td>200</td>
<td>push 200</td>
</tr>
<tr>
<td>1000</td>
<td>push 1000</td>
</tr>
<tr>
<td>vlan-tags outer 2000 inner 300</td>
<td>push 2000, push 300</td>
</tr>
<tr>
<td>vlan-tags outer 100 inner 400</td>
<td>push 100, push 400</td>
</tr>
<tr>
<td>vlan-id-range 10-100</td>
<td>–</td>
</tr>
<tr>
<td>vlan-tags outer 200 inner-range 10-100</td>
<td>–</td>
</tr>
</tbody>
</table>

**Related Documentation**
- Understanding Layer 2 Learning and Forwarding on page 18
- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83

Configuring VLAN Identifiers for Bridge Domains in ACX Series

You can configure VLAN identifiers for a bridge domain for normalization in the following ways:

- Configure VLAN mapping by using the `input-vlan-map` and the `output-vlan-map` statements at the `[edit interfaces interface-name]` hierarchy level.
- Configure an implicit normalizing VLAN identifier under the bridge domain by using the `vlan-id` statement at the `[edit bridge-domains bridge-domain-name]` hierarchy level.

**NOTE:** You cannot configure VLAN mapping by using the `input-vlan-map` and `output-vlan-map` statements if you configure a normalizing VLAN identifier for a bridge domain by using the `vlan-id` statement.

You can use the `vlan-id-list [vlan-id-numbers]` statement to configure bridging for multiple VLANs. Configuring this statement creates a bridge domain for:

- Each VLAN listed—for example, `vlan-id-list [100 200 300]`
- Each VLAN in a range—for example, `vlan-id-list [100-200]`
- Each VLAN in a list and range combination—for example, `vlan-id-list [50, 100-200, 300]`

**Related Documentation**

- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72

Configuring Bridge Domains as Switches for Layer 2 Trunk Ports

You can configure a set of bridge domains that are associated with a Layer 2 trunk port. The set of bridge domains function as a switch. Packets received on a trunk interface are
forwarded within a bridge domain 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 bridge domains, include the following statements:

```plaintext
[edit interfaces]
interface-name {
  unit number {
    family bridge {
      interface-mode access;
      vlan-id number;
    }
  }
}
interface-name {
  native-vlan-id number;
  unit number {
    family bridge {
      interface-mode trunk;
      vlan-id-list [vlan-id-numbers];
    }
  }
}
[edit bridge-domains]
bridge-domain-name {
  vlan-id number;
  vlan-id-list [vlan-id-numbers];
  ....
}
```

For `interface-mode trunk`, you can include the `vlan-id-list` statement.

You must configure a bridge domain 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. For more information about configuring trunk and access interfaces, see the `Interfaces Feature Guide for Security Devices`.

**Related Documentation**
- Configuring a Bridge Domain on page 25
Understanding Layer 2 Virtual Switches

On MX Series routers only, you can group one or more bridge domains to form a virtual switch to isolate a LAN segment with its spanning-tree protocol instance and separate its VLAN ID space. A bridge domain consists of a set of logical ports that share the same flooding or broadcast characteristics. Like a virtual LAN, a bridge domain spans one or more ports of multiple devices. You can configure multiple virtual switches, each of which operates independently of the other virtual switches on the routing platform. Thus, each virtual switch can participate in a different Layer 2 network.

You can configure a virtual switch to participate only in Layer 2 bridging and optionally to perform Layer 3 routing. In addition, you can configure one of three Layer 2 control protocols—Spanning-Tree Protocol, Rapid Spanning-Tree Protocol (RSTP), or Multiple Spanning-Tree Protocol (MSTP)—to prevent forwarding loops. For more information about how to configure Layer 2 logical ports on an interface, see the Junos OS Network Interfaces Library for Routing Devices.

In Junos OS Release 9.2 and later, you can associate one or more logical interfaces configured as trunk interfaces with a virtual switch. A trunk interface, or Layer 2 trunk port, enables you to configure a logical interface to represent multiple VLANs on the physical interface. Packets received on a trunk interface are forwarded within a bridge domain that has same VLAN identifier. For more information about how to configure trunk interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

You can also configure Layer 2 forwarding and learning properties for the virtual switch as well as any bridge domains that belong to a virtual switch.
For more information about configuring a routing instance for Layer 2 VPN, see the Junos OS VPNs Library for Routing Devices.

**Related Documentation**
- Configuring a Layer 2 Control Protocol Routing Instance
- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Layer 2 Protocol Tunneling Through a Network

### Configuring a Layer 2 Virtual 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. Layer 3 routing is not performed. Each bridge domain 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 bridge domains 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:

```text
[edit]
routing-instances {
    routing-instance-name ( instance-type virtual-switch; bridge-domains {
        bridge-domain-name {
            domain-type bridge;
            interface interface-name;
            vlan-id (all | none | number); # Cannot be used with 'vlan-tags' statement
            vlan-id-list [ vlan-id-numbers ];
            vlan-tags outer number inner number; # Cannot be used with 'vlan-id' statement
        }
    }
    protocols {
```
To enable a virtual switch, you must specify `virtual-switch` as the `instance-type`.

For each bridge domain that you configure for the virtual switch, specify a `bridge-domain-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` or `all` options.

The `all` option is not supported with IRB.

**NOTE:** You do not have to specify a VLAN identifier for a bridge domain. However, you cannot specify the same VLAN identifier for more than one bridge domain within a virtual switch. Each bridge domain within a virtual switch must have a unique VLAN identifier.

**NOTE:** For a single bridge domain, you can include either the `vlan-id` statement or the `vlan-tags` statement, but not both. The `vlan-id` statement, `vlan-id-list` statement, and `vlan-tags` statement are mutually exclusive.

The `vlan-id-list` statement allows you to automatically create multiple bridge-domains for each `vlan-id` in the list.

To specify one or more logical interfaces to include in the bridge domain, specify an `interface-name` for an Ethernet interface you configured at the `[edit interfaces]` hierarchy level. For more information, see the Junos OS Network Interfaces Library for Routing Devices.

### Related Documentation
- Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances on page 38
- Configuring Integrated Routing and Bridging for a Bridge Domain in a Layer 2 Virtual Switch on page 50

### Configuring a Virtual Switch Routing Instance on MX Series Routers

On MX Series routers only, use the `virtual-switch` routing instance type to isolate a LAN segment with its spanning-tree instance and to separate its VLAN ID space. A bridge domain consists of a set of ports that share the same flooding or broadcast characteristics. Each virtual switch represents a Layer 2 network. You can optionally configure a virtual switch to support Integrated Routing and Bridging (IRB), which facilitates simultaneous Layer 2 bridging and Layer 3 IP routing on the same interface. You can also configure...
Layer 2 control protocols to provide loop resolution. Protocols supported include the Spanning-Tree Protocol (STP), Rapid Spanning-Tree Protocols (RSTP), Multiple Spanning-Tree Protocol (MSTP), and VLAN Spanning-Tree Protocol (VSTP).

To create a routing instance for a virtual switch, include at least the following statements in the configuration:

```
[edit]
routing-instances {
    routing-instance-name
    instance-type virtual-switch;
    bridge-domains {
        bridge-domain-name {
            domain-type bridge;
            interface interface-name;
            vlan-id (all | none | number);
            vlan-tags outer number inner number;
        }
    }
    protocols {
        (rstp | mstp | vstp) {
            ...stp-configuration ... 
        }
    }
}
```

For more information about configuring virtual switches, see "Configuring a Layer 2 Virtual Switch" on page 48.

Related Documentation

Configuring Integrated Routing and Bridging for a Bridge Domain in a Layer 2 Virtual Switch

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 bridge domain that has a Layer 3 protocol configured. You configure a logical routing interface by including the `irb` statement at `[edit interfaces]` hierarchy level and include that interface in the bridge domain. For more information about how to configure a routing interface, see the Junos OS Network Interfaces Library for Routing Devices.

**NOTE:** You can include only one routing interface in a bridge domain.

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

```
[edit]
routing-instances {
```
To enable a virtual switch, you must specify `virtual-switch` as the `instance-type`. The `instance-type virtual-switch` statement is not supported at the `[edit logical-systems logical-system-name]` hierarchy level.

For each bridge domain that you configure for the virtual switch, specify a `bridge-domain-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:** For a single bridge domain, 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 bridge domain, specify the `interface-name` for each Ethernet interface to include that you configured at the `[edit interfaces irb]` hierarchy level.

To associate a routing interface with a bridge domain, include the `routing-interface routing-interface-name` statement and specify a `routing-interface-name` you configured at the `[edit interfaces irb]` hierarchy level. You can configure only one routing interface for each bridge domain. For more information about how to configure logical and routing interfaces, see the *Junos OS Network Interfaces Library for Routing Devices*.

**NOTE:** If you configure a routing interface to support IRB in a bridge domain, you cannot use the `all` option for the `vlan-id` statement.

**Related Documentation**
- Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances on page 38

**Configuring Integrated Routing and Bridging in ACX Series**

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 bridge domain that has an IRB interface configured. You configure a logical routing interface by including the `irb` statement at the `[edit interfaces]` hierarchy level and include that interface in the bridge domain. For more information about how to configure a routing interface, see the _Junos OS Network Interfaces Library for Routing Devices_.

**NOTE:** You can include only one routing interface in a bridge domain.

The following are the list of features supported for IRB:

- Family `inet`, `inet6`, and `iso` are supported on an IRB interface.
- Routing protocols supported on an IRB interface are BGP, ISIS, OSPF, RIP, IGMP, and PIM.
- DHCP Relay with option 82 is supported on an IRB interface.
- IRB can be added in a VRF routing instance.
- VRRP is supported on an IRB interface.
- Bidirectional Forwarding Detection (BFD) protocol is supported on an IRB interface.
- The following Class-of-Service configurations are supported on an IRB interface:
  - The IRB classifiers and rewrite on routed packets.
  - Fixed classifier can be applied on an IRB logical interface.
  - Firewall filters (multifield filter) can be used to assign forwarding class and loss priority. You should define a family inet or inet6 filter and apply it as the input filter on an IRB logical interface under family inet.

  **NOTE:** `physical-interface-filter` is not supported for family inet6 filter on IRB logical interface.

- Re-write can be applied only at the IRB interface level.
- `dscp`, `inet-precedence`, `ieee-802.1`, and `ieee-802.1ad` values can be rewritten.

ACX routers do not support MPLS families on IRB.

IRB can be configured under the following hierarchies:

- `[edit interfaces irb interface_type]` hierarchy level
  - `disable`—Disables the interface
  - `gratuitous-arp-reply`—Enables gratuitous ARP reply
  - `hold-time`—Hold time for link up and link down
  - `mtu`—Maximum transmit packet size (256..9192)
- no-gratuitous-arp-reply—Does not enable gratuitous ARP reply
- no-gratuitous-arp-request—Ignores gratuitous ARP request

[edit interfaces irb.unit family (inet | inet6 | iso)] hierarchy level

[edit bridge-domains routing-interface interface irb.unit] hierarchy level

[edit routing-instances instance-type interface irb.unit] hierarchy level

[edit protocols (bgp | isis | ospf | rip | igmp | pim) interface irb.unit] hierarchy level

[edit class-of-service interfaces irb]] hierarchy level

In ACX5048 and ACX5096 routers, you can configure IRB at the [edit vlans vlan-name] l3-interface irb.unit; level.

NOTE: The Layer 2 CLI configurations and show commands for ACX5048 and ACX5096 routers differ compared to other ACX Series routers. For more information, see Layer 2 Next Generation Mode for ACX Series.

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

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

For each bridge domain that you configure, specify a `bridge-domain-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.

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

NOTE: For a single bridge domain, 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 bridge domain, specify the `interface-name` for each Ethernet interface to include that you configured at the `[edit interfaces] hierarchy level.
NOTE: A maximum of 4000 active logical interfaces are supported on a bridge domain configured for Layer 2 bridging.

To associate a routing interface with a bridge domain, include the `routing-interface routing-interface-name` statement and specify a `routing-interface-name` you configured at the [edit interfaces irb] hierarchy level. You can configure only one routing interface for each bridge domain. For more information about how to configure logical and routing interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

In Junos OS Release 9.0 and later, IRB interfaces are supported for multicast snooping. For more information about multicast snooping, see the Multicast Protocols Feature Guide.

NOTE: When you configure multiple IRB logical interfaces, all the IRB logical interfaces share the same MAC address.

The following is a sample configuration for IRB over bridge domain:

```
[edit]
interfaces { [edit]
ge-1/0/0 { [edit]
  encapsulation flexible-ethernet-services;
  flexible-vlan-tagging;
  unit 0 { [edit]
    encapsulation vlan-bridge;
    vlan-id 100;
  }
}

g-1/0/1 { [edit]
  encapsulation flexible-ethernet-services;
  flexible-vlan-tagging;
  unit 0 { [edit]
    encapsulation vlan-bridge;
    vlan-id 100;
  }
}

irb { [edit]
  unit 0 { [edit]
    family inet { [edit]
      address 10.0.1.2/24 { [edit]
      }
    }
  }
}

bridge-domains { [edit]
  bd { [edit]
```
Configuring VPLS Ports in a Virtual Switch

In Junos OS Release 9.3 and later, you can configure VPLS ports in a virtual switch so that the logical interfaces of the Layer 2 bridge domains in the virtual switch can handle VPLS routing instance traffic. VPLS configuration no longer requires a dedicated routing instance of type vpls. Packets received on a Layer 2 trunk interface are forwarded within a bridge domain that has the same VLAN identifier.

A trunk interface is implicitly associated with bridge domains based on VLAN membership. Whereas access interfaces can be part of one VLAN only, trunk interfaces multiplex traffic from multiple VLANs and usually interconnect switches. A Layer 2 trunk port also supports IRB.

To configure VPLS ports in a virtual switch, perform the following tasks:

1. To configure the Layer 2 trunk ports that you will associate with the bridge domains in the virtual switch, include the following statements in the configuration:

```plaintext
[edit]
interfaces {
    interface-name {
        unit logical-unit-number { # Call this 'L2-trunk-port-A'
            family bridge {
                interface-mode trunk;
                vlan-id-list [ vlan-id-numbers ]; # Trunk mode VLAN membership for this interface
            }
        }
    }
    .
    .
    .
    interface-name {
        unit logical-unit-number { # Call this 'L2-trunk-port-B'
            family bridge {
                interface-mode trunk;
                vlan-id-list [ vlan-id-numbers ]; # Trunk mode VLAN membership for this interface
            }
        }
    }
}
```
To configure a logical interface as a trunk port, include the `interface-mode` statement and the `trunk` option at the `[edit interfaces interface-name unit logical-unit-number family bridge]` hierarchy level.

To configure all the VLAN identifiers to associate with a Layer 2 trunk port, include the `vlan-id-list [vlan-id-numbers]` statement at the `[edit interfaces interface-name unit logical-unit-number family bridge]` hierarchy level.

Each of the logical interfaces "L2-trunk-port-A" and "L2-trunk-port-B" accepts packets tagged with any VLAN ID specified in the respective `vlan-id-list` statements.

2. To configure a virtual switch consisting of a set of bridge domains that are associated with one or more logical interfaces configured as a trunk ports, include the following statements in the configuration:

```
[edit]
routing-instance {
    routing-instance-name
        instance-type virtual-switch;
        interface L2-trunk-port-A; # Include one trunk port
        interface L2-trunk-port-B; # Include the other trunk port
        bridge-domains {
            bridge-domain-name-0 {
                domain-type bridge;
                vlan-id number;
            }
            bridge-domain-name-1 {
                domain-type bridge;
                vlan-id number;
            }
        }
        protocols {
            vpls {
                vpls-id number;
                ... vpls-configuration ...
            }
        }
    }
}
```

To begin configuring a virtual switch, include the `instance-type` statement and the `virtual-switch` option at the `[edit routing-instances routing-instance-name]` hierarchy level.

To configure a virtual switch consisting of a set of bridge domains that are associated with one or more logical interfaces configured as a trunk ports, you must identify each logical interface by including the `interface interface-name` statement at the `[edit routing-instances routing-instance-name]` hierarchy level.

For each VLAN configured for a trunk port, you must configure a bridge-domain that includes the trunk port logical interface and uses a VLAN identifier within the range carried by that trunk interface. To configure, include the `domain-type bridge, vlan-id`
You can associate one or more Layer 2 trunk interfaces with a virtual switch. A Layer 2 trunk interface enables you to configure a logical interface to represent multiple VLANs on the physical interface. Within the virtual switch, you configure a bridge domain and VLAN identifier for each VLAN identifier configured on the trunk interfaces. Packets received on a trunk interface are forwarded within a bridge domain that has the same VLAN identifier. Each virtual switch you configure operates independently and can participate in a different Layer 2 network.

A virtual switch configured with a Layer 2 trunk port also supports IRB within a bridge domain. 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. For more information about configuring trunk and access interfaces, see the Junos OS Network Interfaces Library for Routing Devices.

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;
        bridge-domains {
            bridge-domain-name {
                vlan-id number;
            }
        }
    }
}
```

**NOTE:** You must configure a bridge domain and VLAN identifier for each VLAN identifier configured for the trunk interface.

Layer 2 trunk ports are used in two distinct types of virtual switch configuration. One method is called service provider style and the other is called enterprise style. The two
methods can be confusing because both methods involve configuring interfaces known as trunk interfaces. However, both types of configuration are distinct.

Service provider style and enterprise style each have benefits and drawbacks.

- **Service provider style**—Offers more control, but requires more care in configuration. Service providers can use all bridging features in any shape or size, but for large bridged designs, customization requirements quickly grow.
- **Enterprise style**—Offers a single Layer 2 network connected by simple bridges. Easier to use, but more limited in function. Configuration is simple and straightforward and condensed.

**NOTE:** The terms “service provider style” and “enterprise style” do not imply any limitations based on organization type or size. Any large enterprise may use service-provider-style configurations and a small regional service provider is free to use enterprise style. The differences apply only to the configuration styles.

The easiest way to understand the differences in configuration of the two styles is to compare them using the same interfaces and VLAN IDs.

You can configure multiple bridge domains between the same pair of Ethernet interfaces, for example, `xe-0/0/1` and `xe-0/0/2`. If there are two bridge domains needed, you can configure one bridge domain as VLAN-100 and the other as VLAN-200. However, the configuration requirements are different when implementing service provider style or enterprise style. Here is a look at both styles using the same interfaces and VLANs.

Service provider style involves configuring the values for three main parameters, plus the bridge domains to connect them:

- **VLAN tagging**—Configure the bridged physical interfaces with `vlan-tagging` to allow them to operate in IEEE 802.1Q mode, also known as a trunk interface.
- **Extended VLAN Bridge**—Configure the physical interface with the encapsulation statement type `extended-vlan-bridge` to allow bridging on each logical interface.
- **Logical unit**—Configure a logical unit for each bridged VLAN ID. In most cases, you configure the unit number to be the same as the VLAN ID (that is, unit 100 = VLAN ID 100).
- **Bridge domains**—Configure the VLAN bridge domains to associate the logical interfaces with the correct VLAN IDs.

Here is the service provider style configuration showing two interfaces used for bridging across two bridge domains, VLAN ID 100 and 200.

```plaintext
[edit]
interfaces {
  xe-0/0/1 {
    vlan-tagging;
  }
}
```
encapsulation extended-vlan-bridge;
unit 100 {
  vlan-id 100;
}
unit 200 {
  vlan-id 200;
}
}
xe-0/0/2 {
  vlan-tagging;
  encapsulation extended-vlan-bridge;
  unit 100 {
    vlan-id 100;
  }
  unit 200 {
    vlan-id 200;
  }
}
}

bridge-domains {
  VLAN-100 {
    vlan-id 100;
    interface xe-0/0/1.100;
    interface xe-0/0/2.100;
  }
  VLAN-200 {
    vlan-id 200;
    interface xe-0/0/1.200;
    interface xe-0/0/2.200;
  }
}

Note that each physical interface has VLAN tagging enabled as well as extended VLAN bridge encapsulation. There are many more parameters that can be configured in service provider style.

In contrast, enterprise style involves configuring the values for three different parameters, plus the bridge domains to connect them:

- Family—Configure each bridged physical interface with the family type **bridge**.
- Interface mode—Configure logical interface so that the physical interface operates as either an untagged access port (not shown in this topic) or as an IEEE 802Q trunk.
- VLAN ID—Configure each logical interface with a VLAN ID to determine with which bridge the interface belongs.
- Bridge domain—Configure the VLAN bridge domains to associate with the correct VLAN IDs.
NOTE: Enterprise style is simpler than the service provider style. Enterprise style automatically places interfaces in bridge domains when the configuration is committed.

Here is the enterprise style configuration showing the same two interfaces used for bridging across the same two bridge domains, VLAN ID 100 and 200.

```
[edit]
interfaces {
  xe-0/0/1 {
    unit 0 {
      family bridge {
        interface-mode trunk;
        vlan-id-list [ 100 200 ];
      }
    }
  }
  xe-0/0/2 {
    unit 0 {
      family bridge {
        interface-mode trunk;
        vlan-id-list [ 100 200 ];
      }
    }
  }
}

bridge-domains {
  VLAN-100 {
    vlan-id 100;
  }
  VLAN-200 {
    vlan-id 200;
  }
}
```

In exchange for simplicity, enterprise style does not allow you to configure VLAN tagging options or encapsulation type. You do not create a separate logical interface for each VLAN ID.

NOTE: You can configure more parameters in each style. These further parameters are beyond the scope of this basic configuration topic.

Related Documentation
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
CHAPTER 4

Configuring Layer 2 Address Learning and Forwarding

- Configuring the MAC Table Timeout Interval on page 61
- Enabling MAC Accounting on page 62
- Limiting the Number of MAC Addresses Learned from Each Logical Interface on page 63
- Disabling Layer 2 Learning and Forwarding on page 64
- Example: Loop Detection Using the MAC Move Approach on page 64

Configuring the MAC Table Timeout Interval

The MAC table aging process ensures that a router tracks only active MAC addresses on the network and is able to flush out addresses that are no longer used.

You can configure the MAC table aging time, the maximum time that an entry can remain in the MAC table before it “ages out,” on all bridge domains, one or all VPLS instances, or one or all Ethernet virtual private network (EVPN) instances on the router. This configuration 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 routing table, the router floods the traffic to all interfaces.

Depending on how long you want to keep a MAC address in a MAC table before it expires, you can either increase or decrease the aging timer. By default, the timeout interval for all entries in the MAC table is 300 seconds. You can modify the timeout interval for MAC table entries on an MX Series router. You cannot modify the timeout interval for a virtual switch.

**NOTE:** The timeout interval applies only to dynamically learned MAC addresses. This value does not apply to configured static MAC addresses, which never time out.

The range for `seconds` is from 10 through 1,000,000.
You can modify the timeout interval for a router (at the global level) or on a per-domain basis (bridge domain).

- To modify the timeout interval for the MAC table for a router:

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

- To modify the timeout interval for a bridge domain:

  ```
  [edit bridge-domain bridge-domain-name bridge-options];
  user@host# set mac-table-aging-time time;
  ```

- To modify the timeout for a VPLS or an Ethernet virtual private network (EVPN) instance within a bridge domain:

  ```
  [edit routing-instance routing-instance-name protocols vpls];
  [edit routing-instance routing-instance-name protocols evpn];
  user@host# set mac-table-aging-time time;
  ```

Related Documentation
- Understanding Layer 2 Learning and Forwarding on page 18
- Enabling MAC Accounting on page 62
- Limiting the Number of MAC Addresses Learned from Each Logical Interface on page 63
- Disabling Layer 2 Learning and Forwarding

**Enabling MAC Accounting**

By default, MAC accounting is disabled. On MX Series routers, you can enable packet accounting either for the router as a whole or for a specific bridge domain. After you enable packet accounting, the Junos OS maintains packet counters for each MAC address learned.

To enable MAC accounting for an MX Series router, include the `global-mac-statistics` statement at the `[edit protocols l2-learning]` hierarchy level:

```
[edit protocols l2-learning]
global-mac-statistics;
```
Limiting the Number of MAC Addresses Learned from Each Logical Interface

You can configure a limit to the number of MAC addresses learned from the logical interfaces on an MX Series router.

To configure a limit to the total number of MAC addresses that can be learned from the logical interfaces, include the `global-mac-limit limit` statement at the `[edit protocols l2-learning] hierarchy level:

```plaintext
[edit]
protocols {
    l2-learning {
        global-mac-limit limit;
    }
}
```

The default limit to the number of MAC addresses that can be learned the router as a whole is 393,215. The range that you can configure for the router as a whole is 20 through 1,048,575.

After the configured MAC address limit is reached, the default is for packets to be forwarded. You can specify that the packets be dropped by including the `packet-action drop` statement at the `[edit protocols l2-learning global-mac-limit] hierarchy level:

```plaintext
[edit]
protocols {
    l2-learning {
        global-mac-limit limit {
            packet-action drop;
        }
    }
}
```

You can also configure a limit to the number of MAC address learned from all the interfaces in a bridge domain or from a specific logical interface only.

---

**NOTE:** The behavior is different for some configurations. For aggregated Ethernet interfaces and label-switched interfaces, the behavior is to learn all the new MAC addresses even when the limit has been reached. The excess addresses are later deleted. The learning limit does not apply to bridge domain trunk ports, because they have no counters for the individual domains, and those domains might have different MAC learning limits.

---

**NOTE:** When static MAC addresses are configured, the learning limit is the configured limit minus the number of static addresses.
NOTE: On MX Series routers running Junos OS Release 8.4 and later, statistics for an aged destination MAC entry are not retained. In addition, source and destination statistics are reset during a MAC move. In previous releases, only source statistics were reset during a MAC move.

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:

```plaintext
[edit protocols l2-learning]
global-no-mac-learning;
```

For information about how to configure a virtual switch, see “Configuring a Layer 2 Virtual Switch” on page 48.

Related Documentation
- Understanding Layer 2 Learning and Forwarding on page 18
- Configuring the MAC Table Timeout Interval on page 61
- Enabling MAC Accounting on page 62
- Limiting the Number of MAC Addresses Learned from Each Logical Interface on page 63

Example: Loop Detection Using the MAC Move Approach

This example shows how to detect loops using the MAC move approach.

- Requirements on page 64
- Overview on page 65
- Configuration on page 65
- Verification on page 67

Requirements

This example requires the following hardware and software components:

- MX Series 5G Universal Routing Platforms
- Junos OS Release 13.2 running on all the devices
Overview

When a MAC address appears on a different physical interface or within a different unit of the same physical interface and if this behavior occurs frequently, it is considered a MAC move.

Configuration errors at the network can force traffic into never ending circular paths. Once there are loops in the Layer 2 network, one of the symptoms is frequent MAC moves, which can be used for rectification of the problem. When it is observed that a source MAC address is moving among the ports, interface is blocked based on the configured action-priority for the interface. If the action-priority value configured for interfaces is the same, the last interface for the bridge domain on which the MAC address move occurred is blocked.

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, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set interfaces ge-1/0/4 vlan-tagging
set interfaces ge-1/0/4 encapsulation flexible-ethernet-services
set interfaces ge-1/0/4 unit 10 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 10 vlan-id 10
set interfaces ge-1/0/4 unit 11 encapsulation vlan-bridge
set interfaces ge-1/0/4 unit 11 vlan-id 11
set interfaces ge-1/0/5 unit 0 family bridge interface-mode trunk
set interfaces ge-1/0/5 unit 0 family bridge vlan-id-list 10-12
set interfaces ge-1/0/6 unit 0 family bridge interface-mode trunk
set interfaces ge-1/0/6 unit 0 family bridge vlan-id-list 10-12
set bridge-domains bd10 vlan-id 10
set bridge-domains bd10 enable-mac-move-action
set bridge-domains bd10 bridge-options interface ge-1/0/5.0 action-priority 1
set bridge-domains bd10 bridge-options interface ge-1/0/6.0 action-priority 5
set bridge-domains bd11 vlan-id 11
set bridge-domains bd11 enable-mac-move-action
set bridge-domains bd12 vlan-id 12
```

In the previous example, all the interfaces, including the trunk interfaces in bd10 and bd11 will be monitored. If there are frequent MAC moves detected within interfaces ge-1/0/5 and ge-1/0/6, interface ge-1/0/5 is blocked. The blocking for trunk interfaces is such that data traffic only for a VLAN (on which the MAC move is detected) will be blocked and not for all the VLANs in the trunk. No action will be taken if a frequent MAC move is observed in bd12.
Configuring Loop Detection Using the MAC Move Approach

**Step-by-Step Procedure**

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 loop detection using the MAC address move approach:

1. Configure the interfaces.

   ```
   [edit interfaces]
   user@host# set ge-1/0/4 vlan-tagging
   user@host# set ge-1/0/4 encapsulation flexible-ethernet-services
   user@host# set ge-1/0/4 unit 10 encapsulation vlan-bridge
   user@host# set ge-1/0/4 unit 10 vlan-id 10
   user@host# set ge-1/0/4 unit 11 encapsulation vlan-bridge
   user@host# set ge-1/0/4 unit 11 vlan-id 11
   user@host# set ge-1/0/5 unit 0 family bridge interface-mode trunk
   user@host# set ge-1/0/5 unit 0 family bridge vlan-id-list 10-12
   user@host# set ge-1/0/6 unit 0 family bridge interface-mode trunk
   user@host# set ge-1/0/6 unit 0 family bridge vlan-id-list 10-12
   ```

2. Configure the bridge domain parameters.

   ```
   [edit bridge-domains]
   user@host# set bd10 vlan-id 10
   user@host# set bd10 enable-mac-move-action
   user@host# set bd10 bridge-options interface ge-1/0/5.0 action-priority 1
   user@host# set bd10 bridge-options interface ge-1/0/6.0 action-priority 5
   user@host# set bd11 vlan-id 11
   user@host# set bd11 enable-mac-move-action
   user@host# set bd12 vlan-id 12
   ```

**Results**

From configuration mode, confirm your configuration by entering `show interfaces` and `show bridge-domains` commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
ge-1/0/4 {
  vlan-tagging;
  encapsulation flexible-ethernet-services;
  unit 10 {
```
encapsulation vlan-bridge;
  vlan-id 10;
}
unit 11 {
  encapsulation vlan-bridge;
  vlan-id 11;
}
  ge-1/0/5 {
    unit 0 {
      family bridge {
        interface-mode trunk;
        vlan-id-list 10-12;
      }
    }
  }
  ge-1/0/6 {
    unit 0 {
      family bridge {
        interface-mode trunk;
        vlan-id-list 10-12;
      }
    }
  }

user@host# show bridge-domains
bridge-domains {
  bd10 {
    vlan-id 10;
    bridge-options {
      interface ge-1/0/5.0 {
        action-priority 1;
      }
      interface ge-1/0/6.0 {
        action-priority 5
      }
    }
    enable-mac-move-action;
  }
  bd11 {
    vlan-id 11;
    enable-mac-move-action;
  }
  bd12 {
    vlan-id 12;
  }
}

If you are done configuring the device, enter commit from configuration mode.

**Verification**

**Verifying That the Logical Interfaces Blocked Due to MAC Move Are Displayed**

**Purpose** Ensure that the current set of logical interfaces blocked due to a MAC move, if any, are displayed.
**Action**  
From operational mode, enter the `show l2-learning mac-move-buffer active` command.

```
user@host# show l2-learning mac-move-buffer active
MAC Address: 00:00:00:00:01:01, VLAN Id: 0
Time Rec : 2012-06-25 06:23:41   Bridge Domain: bd10
Prev IFL : ge-1/0/5.0            New IFL: ge-1/0/6.0
IFBD     : ge-1/0/6.0:10         Blocked : YES
```

**Meaning**  
As a result of MAC move detection, one of the involved interface bridge domains will be blocked. The output shows that the ge-1/0/6 logical interface is blocked.

**Related Documentation**
- `bridge-domains on page 99`
- `Understanding Layer 2 Learning and Forwarding on page 18`
CHAPTER 5

Configuring Layer 2 Learning and Forwarding for Bridge Domains

- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain on page 70
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for a Bridge Domain on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72
- Limiting MAC Addresses Learned from an Interface in a Bridge Domain on page 73
- Configuring MAC Address Limits on a Logical Interface on page 75
- Enabling MAC Accounting for a Bridge Domain on page 78
- Disabling MAC Learning for a Bridge Domain or Logical Interface on page 78
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Preventing Communication Among Customer Edge Devices as ACX Routers on page 80

Understanding Layer 2 Learning and Forwarding for Bridge Domains

When you configure a bridge domain, Layer 2 address learning is enabled by default. The bridge domain learns unicast media access control (MAC) addresses to avoid flooding the packets to all the ports in the bridge domain. Each bridge domain 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 bridge domain.

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 router or for a specific bridge domain or logical interface. You can also configure the following Layer 2 learning and forwarding properties:
• 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 bridge domain
• Size of the MAC address table for the bridge domain
• MAC accounting for a bridge domain

Related Documentation
• Understanding Layer 2 Learning and Forwarding on page 18

Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain

You can manually add static MAC entries for the logical interfaces in a bridge domain. 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 bridge domain, include the static-mac mac-address statement at the [edit bridge-domains bridge-domain-name bridge-options interface interface-name] hierarchy level.

```
[edit]
bridge-domains {
    bridge-domain-name {
        domain-type bridge;
        bridge-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 bridge domain.

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 Bridge Domain or Logical Interface on page 78
• Configuring the Size of the MAC Address Table for a Bridge Domain on page 71
• Limiting MAC Addresses Learned from an Interface in a Bridge Domain on page 73
• Enabling MAC Accounting for a Bridge Domain on page 78
Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series

You can manually add static MAC entries for the logical interfaces in a bridge domain. 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 bridge domain, include the `static-mac mac-address` statement at the `[edit bridge-domains bridge-domain-name bridge-options interface interface-name]` hierarchy level.

```
[edit]
bridge-domains {
  bridge-domain-name {
    bridge-options {
      interface interface-name {
        static-mac mac-address {
        }
      }
    }
  }
}
```

Related Documentation
- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72

Configuring the Size of the MAC Address Table for a Bridge Domain

You can modify the size of the MAC address table for each bridge domain. 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 bridge-domains bridge-domain-name bridge-options]` hierarchy level:

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

**Related Documentation**
- Disabling MAC Learning for a Bridge Domain or Logical Interface on page 78
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain on page 70
- Limiting MAC Addresses Learned from an Interface in a Bridge Domain on page 73
- Enabling MAC Accounting for a Bridge Domain on page 78

**Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series**

You can modify the size of the MAC address table for each bridge domain. The default table size is 5120 addresses per bridge domain. The minimum you can configure is 1 address, and the maximum is 32,000 addresses.

If the MAC table limit is reached, new addresses can no longer be added to the table.

**NOTE:** Unused MAC addresses are removed from the MAC address table automatically. This frees space in the table thereby allowing new entries to be added.

To modify the size of the MAC table, include the `mac-table-size limit` statement at the `[edit bridge-domains bridge-domain-name bridge-options]` hierarchy level:

```
[edit]
bridge-domains {
  bridge-domain-name {
    bridge-options {
      mac-table-size limit {
        packet-action drop;
      }
    }
  }
}
```
NOTE: The mac-table-size CLI statement is not supported on ACX5048 and ACX5096 routers.

Related Documentation

- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71

Limiting MAC Addresses Learned from an Interface in a Bridge Domain

You can configure a limit on the number of MAC addresses learned from a specific bridge domain or from a specific logical interface that belongs to a bridge domain.

To configure a limit for the number of MAC addresses learned from each logical interface in a bridge domain, include the interface-mac-limit limit statement at the [edit bridge-domains bridge-domain-name bridge-options] hierarchy level:

```
[edit]
bridge-domains {
    bridge-domain-name {
        domain-type bridge;
        interface interface-name;
        bridge-options {
            interface-mac-limit limit;
        }
    }
}
```

To limit the number of MAC addresses learned from a specific logical interface in a bridge domain or an entire bridge domain, include the interface-mac-limit limit statement at the [edit bridge-domains bridge-domain-name bridge-options interface interface-name] or [edit bridge-domains bridge-domain-name bridge-options] hierarchy level:

```
[edit]
bridge-domains {
    bridge-domain-name {
        domain-type bridge;
        interface interface-name;
        bridge-options {
            interface-mac-limit limit
        }
    }
}
```
For an access port, the default limit on the maximum number of MAC addresses that can be learned on an access port is 1024. Because an access port can be configured in only one bridge domain in a network topology, the default limit is 1024 addresses, which is same as the limit for MAC addresses learned on a logical interface in a bridge domain (configured by including the `interface-mac-limit limit` statement at the `edit bridge-domains bridge-domain-name bridge-options interface interface-name` or `edit bridge-domains bridge-domain-name bridge-options`) hierarchy level.

For a trunk port, the default limit on the maximum number of MAC addresses that can be learned on a trunk port is 8192. Because a trunk port can be associated with multiple bridge domains, the default limit is the same as the limit for MAC addresses learned on a logical interface in a virtual switch instance (configured by including the `interface-mac-limit limit` statement at the `edit routing-instances routing-instance-name switch-options interface interface-name` for a virtual switch instance).

The value you configure for a specific logical interface overrides any value you specify for the entire bridge domain at the `edit bridge-domains bridge-domain-name bridge-options` hierarchy level.

The default limit to the number of MAC addresses that can be learned on a logical interface is 1024. The range that you can configure for a specific logical interface is 1 through 131,071.

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 by including the `packet-action drop` statement. To specify that packets be dropped for the entire bridge domain, include the `packet-action drop` statement at the `[edit bridge-domains bridge-domain-name bridge-options interface-mac-limit limit]` hierarchy level:

```plaintext
packet-action drop;
```

To specify that the packets be dropped for a specific logical interface in a bridge domain, include the `packet-action drop` statement at the `[edit bridge-domains bridge-domain-name bridge-options interface interface-name interface-mac-limit limit]` hierarchy level:

```plaintext
[edit bridge-domains bridge-domain-name bridge-options interface interface-name interface-mac-limit limit]
packet-action drop;
```
NOTE: The behavior is different for some configurations. For aggregated Ethernet interfaces and label-switched interfaces, the behavior is to learn all the new MAC addresses even when the limit has been reached. The excess addresses are later deleted. The learning limit does not apply to bridge domain trunk ports, because they have no counters for the individual domains, and those domains might have different MAC learning limits.

NOTE: When static MAC addresses are configured, the learning limit is the configured limit minus the number of static addresses.

NOTE:

You can also configure a limit to the number of MAC addresses learned for an MX Series router.

Related Documentation

- Disabling MAC Learning for a Bridge Domain or Logical Interface on page 78
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain on page 70
- Configuring the Size of the MAC Address Table for a Bridge Domain on page 71
- Enabling MAC Accounting for a Bridge Domain on page 78

Configuring MAC Address Limits on a Logical Interface

You can configure a limit on the number of MAC addresses learned from a specific logical interface. This feature allows the MAC address table space to be distributed among different logical interfaces, thereby avoiding congestion. The MAC address limit can be applied for both VLAN and VPLS routing instances and by default the MAC limit depends on the profile configured. You can limit the number of MAC addresses learned for a bridge domain and a logical interface at the same time.

NOTE: MAC address limiting is supported only on ACX5000 line of routers.

- Configuring MAC Address Limit on page 75
- Configuring MAC Address Limit for VLANs on page 76
- Configuring MAC Address Limit for VPLS on page 76
- CLI Commands to Configure MAC Address Limiting on page 77

Configuring MAC Address Limit

You can configure the MAC Address limit by using the `set protocols l2-learning global-no-hw-mac-learning` CLI command.
The following configuration example enables limiting MAC address learning on logical interfaces:

```
[edit protocols]
l2-learning {
  global-no-hw-mac-learning;
}
```

### Configuring MAC Address Limit for VLANs

To configure a limit for the number of MAC addresses learned on each logical interface in a VLAN, include the `interface-mac-limit limit` statement at the `[edit vlans vlan-name]` hierarchy level. To limit the MAC addresses learned on a specific logical interface of the VLAN, include the `interface-mac-limit limit` statement at the `[edit vlans vlan-name interface interface-name]` hierarchy level. To limit the MAC addresses learned on each of the logical interfaces of the VLAN, include the `interface-mac-limit limit` statement at the `[edit vlans vlan-name switch-options]` hierarchy level.

The following example configures a limit for the number of MAC addresses learned on a logical interface in a VLAN:

```
[edit vlans]
vlan10 {
  interface ge-0/0/3.1;
  interface ge-0/0/1.5;
  switch-options {
    interface-mac-limit {
      10;
    }
  }
  interface ge-0/0/1.5 {
    interface-mac-limit {
      20;
    }
  }
}
```

### Configuring MAC Address Limit for VPLS

To configure a limit for the number of MAC addresses learned on each logical interface in a VPLS routing instance, include the `interface-mac-limit limit` statement at the `[edit routing-instances routing-instance-name protocols vpls]` hierarchy level. To limit the MAC addresses learned on a specific logical interface of the VPLS instance, include the `interface-mac-limit limit` statement at the `[edit routing-instances routing-instance-name protocols vpls interface interface-name]` hierarchy level.

The following is an example to configure a limit for the number of MAC addresses learned on a logical interface in VPLS routing instance:

```
[edit routing-instance]
v1 {
  protocols {
```
If you have configured an interface MAC address limit for the logical interface in a bridge domain and a global MAC address limit for a bridge domain, then the interface MAC address limit is considered. The following example shows two MAC address limits configured on the interface ge-0/0/3.5 with the global value as 50 and local value as 30. In this case, the MAC address limit of 30 is considered for the interface ge-0/0/3.5 in the bridge domain.

### CLI Commands to Configure MAC Address Limiting

The following CLI commands are used for configuring MAC address limiting:

- **set protocols l2-learning global-no-hw-mac-learning**—Command to change the hardware-based MAC learning to software-based MAC learning mode.

- **set vlans vlan-name switch-options interface-mac-limit limit**—Command to configure the MAC address limit for each logical interface in a VLAN. The limit is applied to all logical interfaces belonging to the VLAN for which a separate interface MAC address limit is not configured.

- **set vlans vlan-name switch-options interface interface-name interface-mac-limit limit**—Command to configure the interface MAC address limit for a logical interface in a VLAN. The limit is applied to a specific logical interface in the VLAN for which it is configured.
- **set routing-instances routing-instance-name protocols vpls interface-mac-limit limit**—Command to configure the MAC address limit for each logical interface in the VPLS routing instance. This limit is applied to all logical interfaces belonging to the VPLS for which a separate interface MAC address limit is not configured.

- **set routing-instances routing-instance-name protocols vpls interface interface-name interface-mac-limit limit**—Command to configure the interface MAC address limit for a logical interface in the VPLS. This limit is applied to a specific logical interface in the VPLS for which it is configured.

**Related Documentation**

**Enabling MAC Accounting for a Bridge Domain**

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

To enable MAC accounting for a bridge domain, include the `mac-statistics` statement at the `[edit bridge-domains bridge-domain-name bridge-options]` hierarchy level:

```
[edit bridge-domains bridge-domain-name bridge-options]
mac-statistics;
```

**Related Documentation**

- Disabling MAC Learning for a Bridge Domain or Logical Interface on page 78
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain on page 70
- Configuring the Size of the MAC Address Table for a Bridge Domain on page 71
- Limiting MAC Addresses Learned from an Interface in a Bridge Domain on page 73

**Disabling MAC Learning for a Bridge Domain or Logical Interface**

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

To disable MAC learning for all logical interfaces in a bridge domain in a virtual switch, include the `no-mac-learning` statement at the `[edit bridge-domains bridge-domain-name bridge-options]` hierarchy level:

```
[edit]
bridge-domains {
  bridge-domain-name {
    domain-type bridge;
    interface interface-name;
    bridge-options {
      no-mac-learning;
    }
  }
}
```
To disable MAC learning for a specific logical interface in a bridge domain, include the `no-mac-learning` statement at the `[edit bridge-domains bridge-domain-name bridge-options interface interface-name]` hierarchy level.

```
[edit]
bridge-domains {
  bridge-domain-name {
    domain-type bridge;
    interface interface-name;
    bridge-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 bridge domain.

**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.

**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.

### Related Documentation
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain on page 70
- Configuring the Size of the MAC Address Table for a Bridge Domain on page 71
- Limiting MAC Addresses Learned from an Interface in a Bridge Domain on page 73
- Enabling MAC Accounting for a Bridge Domain on page 78

### Disabling MAC Learning for Bridge Domains on ACX Series

You can disable MAC learning in a bridge domain. Disabling dynamic MAC learning prevents the bridge domain from learning source MAC addresses.
To disable MAC learning in a bridge domain, include the `no-mac-learning` statement at the `[edit bridge-domains bridge-domain-name bridge-options]` hierarchy level:

```
[edit]
bridge-domains {
  bridge-domain-name {
    bridge-options {
      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 bridge domain.

### Related Documentation
- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72

### Preventing Communication Among Customer Edge Devices as ACX Routers

In a bridge domain, when a frame is received from a CE interface, it is flooded to the other CE interfaces and all of the provider edge (PE) interfaces if the destination MAC address is not learned or if the frame is either broadcast or multicast. If the destination MAC address is learned on another CE device, such a frame is unicasted to the CE interface on which the MAC address is learned. This might not be desirable if the service provider does not want CE devices to communicate with each other directly.

To prevent CE devices from communicating directly, include the `no-local-switching` statement at the `[edit bridge-domains bridge-domain-name]` hierarchy level. Configure the logical interfaces in the bridge domain as core-facing (PE interfaces) by including the `core-facing` statement at the `[edit interfaces interface-name unit logical-unit-number family family]` hierarchy level to specify that the VLAN is physically connected to a core-facing ISP router and ensures that the network does not improperly treat the interface as a client interface. When specified, traffic from one CE interface is not forwarded to another CE interface.
For the **no-local-switching** option, integrated routing and bridging (IRB) configured on a bridge domain with this option enabled is not treated as a designated CE or PE interface. Traffic arriving from a CE or PE interface can navigate towards IRB and traffic that reaches in the input direction to the IRB can pass out of a CE or PE interface. The disabling of local switching achieves the functionality of split-horizon in a bridge domain. If no-local-switching is configured in a bridge domain, then traffic cannot flow between CE and CE interfaces. This stoppage of traffic flow includes known unicast and multicast, unknown unicast and multicast, and broadcast traffic. However, traffic continues to be transmitted between CE and PE interfaces, and PE and PE interfaces.

**Related Documentation**
- **no-local-switching**
Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports

Layer 2 learning is enabled by default. A set of bridge domains, 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 bridge domains 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 bridge domains
- Modify the size of the MAC address table for the set of bridge domains
- Enable MAC accounting for the set of bridge domains
Limiting MAC Addresses Learned from a Layer 2 Trunk Port

You can configure a limit on the number of MAC addresses learned from a trunk port or from a specific trunk or access interface.

To limit the number of MAC addresses learned through a trunk port associated with a set of bridge domains, include the `interface-mac-limit limit` statement at the `[edit switch-options]` hierarchy level:

```
[edit]
switch-options {
    interface-mac-limit limit;
}
```

To limit the number of MAC addresses learned from a specific logical interface configured as an access interface or a trunk interface, include the `interface-mac-limit limit` statement at the `[edit switch-options interface interface-name]` hierarchy level:

```
[edit]
switch-options {
    interface interface-name {
        interface-mac-limit limit;
    }
}
```

The default value for the number MAC addresses that can be learned from a logical interface is 1024. You can specify a limit either for a set of bridge domains or for a specific logical interface in the range from 1 through 131,071. The value you configure for a specific logical interface overrides any value you specify for the set of bridge domains.

After the specified 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 for the entire virtual switch after the MAC address limit is reached by including the `packet-action drop` statement at the `[edit switch-options interface-mac-limit limit]` hierarchy level:

```
[edit switch-options interface interface-name interface-mac-limit limit]
packet-action drop;
```

To specify that the packets be dropped from a specific logical interface in a set of bridge domains with a trunk port after the MAC address limit is reached, include the `packet-action drop` statement at the `[edit routing-instances routing-instance-name interface interface-name interface-mac-limit limit]` hierarchy level:

```
[edit routing-instances routing-instance-name interface interface-name interface-mac-limit limit]
packet-action drop;
```
Configuring the Size of the MAC Address Table for a Set of Bridge Domains

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

To modify the size of the MAC table for a set of bridge domains, include the `mac-table-size` statement at the `[edit switch-options]` hierarchy level:

```
[edit switch-options]
mac-table-size limit;
```

Related Documentation
- Disabling MAC Learning for a Set of Bridge Domains on page 86
- Configuring the Size of the MAC Address Table for a Set of Bridge Domains on page 85
- Enabling MAC Accounting for a Set of Bridge Domains on page 85

Enabling MAC Accounting for a Set of Bridge Domains

By default, MAC accounting is disabled. You can enable packet counting for a set of bridge domains. 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 bridge domains.

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

```
[edit switch-options]
mac-statistics;
```

Related Documentation
- Disabling MAC Learning for a Set of Bridge Domains on page 86
- Limiting MAC Addresses Learned from a Layer 2 Trunk Port on page 84
- Enabling MAC Accounting for a Set of Bridge Domains on page 85
Disabling MAC Learning for a Set of Bridge Domains

By default, MAC learning is enabled for a set of bridge domains. You can disable MAC learning for a set of bridge domains. Disabling dynamic MAC learning prevents the Layer 2 trunk port associated with the set of bridge domains 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 bridge domains, include the `no-mac-learning` statement at the `[edit switch-options]` hierarchy level:

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

Related Documentation

- Limiting MAC Addresses Learned from a Layer 2 Trunk Port on page 84
- Configuring the Size of the MAC Address Table for a Set of Bridge Domains on page 85
- Enabling MAC Accounting for a Set of Bridge Domains on page 85
CHAPTER 7

Configuring Q-in-Q Tunneling

- Q-in-Q Tunneling on ACX Series Overview on page 87
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Example: Interconnecting Customer Sites Using Q-in-Q Tunneling on page 89

Q-in-Q Tunneling on ACX Series Overview

Q-in-Q tunneling allows 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. Service providers can use Q-in-Q tunneling to isolate customer traffic within a single site or to enable customer traffic flows across 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.

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

In ACX Series routers, you can configure Q-in-Q tunneling by explicitly configuring an input VLAN map with push function on customer facing interfaces in a bridge domain.

You can configure Q-in-Q tunneling on aggregated Ethernet interface by configuring input and output VLAN map.

Related Documentation
- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring Q-in-Q Tunneling on ACX Series on page 88
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
Configuring Q-in-Q Tunneling on ACX Series

To configure Q-in-Q tunneling, you need to configure the logical interface connected to the customer network (user-to-network interfaces (UNI)) and the logical interface connected to the service provider network (network-to-network interface (NNI)).

The following is an example to configure a logical interface connected to a customer network:

```
[edit]
interface ge-1/0/1 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
        vlan-id-list 10-20;
        input-vlan-map {
            push;
            vlan-id 500;
        }
        output-vlan-map pop;
    }
}
```

The following is an example to configure a logical interface connected to a service provider network:

```
[edit]
interface ge-1/0/2 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
    unit 0 {
        encapsulation vlan-bridge;
        vlan-id 500;
    }
}
```

The following is an example to configure the bridge domain:

```
[edit]
bridge-domains {
    qnq-stag-500 {
        interface ge-1/0/1;
        interface ge-1/0/2;
    }
```

```
You can configure Q-in-Q tunneling on aggregated Ethernet interface connected to the customer network (UNI) and the logical interface connected to the service provider network (NNI).

### Related Documentation
- Layer 2 Bridge Domains on ACX Series Overview on page 19
- Q-in-Q Tunneling on ACX Series Overview on page 87
- Layer 2 Learning and Forwarding for Bridge Domains Overview on page 22
- Configuring a Bridge Domain on ACX Series Routers on page 28
- Configuring VLAN Identifiers for Bridge Domains in ACX Series on page 44
- Disabling MAC Learning for Bridge Domains on ACX Series on page 79
- Configuring Static MAC Addresses for Logical Interfaces in a Bridge Domain in ACX Series on page 71
- Configuring the Size of the MAC Address Table for Bridge Domains in ACX Series on page 72

### Example: Interconnecting Customer Sites Using Q-in-Q Tunneling

This example shows how to interconnect two customer sites (E-LINE Services) based on Q-in-Q tunneling.

- Requirements on page 89
- Overview on page 89
- Configuration on page 90
- Verification on page 93

### Requirements

This example requires the following hardware and software components:

- Three ACX5448 routers configured as PE routers
- Junos OS Release 18.2 or later

### Overview

This example uses ACX5448 routers as PE routers to interconnect two customer sites (Headquarters and Branch offices) using Q-in-Q tunneling.

The configuration shows PE1 (ACX5448) router connected to the Headquarters and PE2 (ACX5448) and PE3 (ACX5448) routers connected to two branch offices (A and B), respectively. The CE facing interface, connected to the Headquarters, is configured to allow CVLAN and add push SVLAN, transported through a service provider/core network.
Configuration

CLI Quick Configuration

The following configuration shows:

- Logical interfaces (Q-in-Q tunnelling) on the PE1, PE2, and PE3 routers connected to CE and service provider.
- VLANs (CVLAN and SVLAN)

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, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

Device PE1

```
user@host# set interfaces ge-1/0/1 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/1 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/1 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/1 unit 0 vlan-id-list 100-200;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map push;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map vlan-id 300;
user@host# set interfaces ge-1/0/1 unit 0 output-vlan-map pop;
user@host# set interfaces ge-1/0/2 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/2 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/2 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/2 unit 0 vlan-id 300;
user@host# set vlan qnq-vlan-300 interface ge-1/0/1;
user@host# set vlan qnq-vlan-300 interface ge-1/0/2;
```

Device PE2

```
user@host# set interfaces ge-1/0/1 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/1 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/1 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 200;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map push;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map vlan-id 300;
user@host# set interfaces ge-1/0/1 unit 0 output-vlan-map pop;
user@host# set interfaces ge-1/0/2 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/2 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/2 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/2 unit 0 vlan-id 300;
user@host# set vlan qnq-vlan-300 interface ge-1/0/1;
user@host# set vlan qnq-vlan-300 interface ge-1/0/2;
```

Device PE3

```
user@host# set interfaces ge-1/0/1 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/1 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/1 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 100;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map push;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map vlan-id 300;
user@host# set interfaces ge-1/0/1 unit 0 output-vlan-map pop;
user@host# set interfaces ge-1/0/2 flexible-vlan-tagging;
```
Configuring the device PE1

Step-by-Step Procedure

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.

1. Configure logical interface (Q-in-Q tunneling) on the PE1 router connected to CE (Headquarters):

   ```
   [edit]
   user@host# set interfaces ge-1/0/1 flexible-vlan-tagging;
   user@host# set interfaces ge-1/0/1 encapsulation flexible-ethernet-services;
   user@host# set interfaces ge-1/0/1 unit 0 encapsulation vlan-bridge;
   user@host# set interfaces ge-1/0/1 unit 0 vlan-id-list 100-200;
   user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map push;
   user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map vlan-id 300;
   user@host# set interfaces ge-1/0/1 unit 0 output-vlan-map pop;
   ```

2. Configure logical interface on the PE1 router connected to a service provider/core network:

   ```
   [edit]
   user@host# set interfaces ge-1/0/2 flexible-vlan-tagging;
   user@host# set interfaces ge-1/0/2 encapsulation flexible-ethernet-services;
   user@host# set interfaces ge-1/0/2 unit 0 encapsulation vlan-bridge;
   user@host# set interfaces ge-1/0/2 unit 0 vlan-id 300;
   ```

3. Configure a VLAN for SVLAN:

   ```
   [edit]
   user@host# set vlan qnq-vlan-300 interface ge-1/0/1;
   user@host# set vlan qnq-vlan-300 interface ge-1/0/2;
   ```

Configuring the device PE2

Step-by-Step Procedure

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.

1. Configure logical interface (Q-in-Q tunneling) on the PE2 router connected to CE (Branch Office A):

   ```
   user@host# set interfaces ge-1/0/2 encapsulation flexible-ethernet-services;
   user@host# set interfaces ge-1/0/2 unit 0 encapsulation vlan-bridge;
   user@host# set interfaces ge-1/0/2 unit 0 vlan-id 300;
   user@host# set vlan qnq-vlan-300 interface ge-1/0/1;
   user@host# set vlan qnq-vlan-300 interface ge-1/0/2;
   ```
2. Configure logical interface on the PE2 router connected to a service provider/core network:

```plaintext
[edit]
user@host# set interfaces ge-1/0/2 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/2 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/2 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/2 unit 0 vlan-id 300;
```

3. Configure a VLAN for SVLAN:

```plaintext
[edit]
user@host# set vlan qnq-vlan-300 interface ge-1/0/1;
user@host# set vlan qnq-vlan-300 interface ge-1/0/2;
```

**Configuring the device PE3**

**Step-by-Step Procedure**

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*.

1. Configure logical interface (Q-in-Q tunneling) on the PE3 router connected to CE (Branch Office B):

```plaintext
[edit]
user@host# set interfaces ge-1/0/1 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/1 encapsulation flexible-ethernet-services;
user@host# set interfaces ge-1/0/1 unit 0 encapsulation vlan-bridge;
user@host# set interfaces ge-1/0/1 unit 0 vlan-id 100;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map push;
user@host# set interfaces ge-1/0/1 unit 0 input-vlan-map vlan-id 300;
user@host# set interfaces ge-1/0/1 unit 0 output-vlan-map pop;
```

2. Configure logical interface on the PE3 router connected to a service provider/core network:

```plaintext
[edit]
user@host# set interfaces ge-1/0/2 flexible-vlan-tagging;
user@host# set interfaces ge-1/0/2 encapsulation flexible-ethernet-services;
```

[Copyright © 2018, Juniper Networks, Inc.]
3. Configure a VLAN for SVLAN:

```plaintext
[edit]
user@host# set vlan qnq-vlan-300 interface ge-1/0/1;
user@host# set vlan qnq-vlan-300 interface ge-1/0/2;
```

**Verification**

**Verifying That the VLAN (bridge domain) Has Been Configured Properly**

**Purpose**  Ensure that the VLAN (bridge domain) has been configured properly.

**Action**  From operational mode, enter the `show vlans` command.

```plaintext
user@host# show vlans
```

<table>
<thead>
<tr>
<th>Routing instance</th>
<th>VLAN name</th>
<th>Tag</th>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>default-switch</td>
<td>default</td>
<td>1</td>
<td>ge-1/0/2</td>
</tr>
<tr>
<td>default-switch</td>
<td>qnq-vlan-300</td>
<td>300</td>
<td>ge-1/0/1.0 ge-1/0/2.0</td>
</tr>
</tbody>
</table>

**Verifying That the MAC Address is Learnt at the PE1 Port (Headquarters)**

**Purpose**  Ensure that the MAC Address is learnt at the PE1 port (Headquarters).
### Purpose

Ensure that the MAC Address is learnt at the PE2 port (Branch Office A).

### Action

From operational mode, enter the **show ethernet-switching table** command.

```
user@host# show ethernet-switching table
```

**Verifying That the MAC Address is Learnt at the PE2 Port (Branch Office A)**

<table>
<thead>
<tr>
<th>Vlan name</th>
<th>MAC address</th>
<th>MAC flags</th>
<th>Age</th>
<th>Logical interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>qnn-vlan-300</td>
<td>b0:c6:9a:ca:3c:01</td>
<td>D</td>
<td>-</td>
<td>ge-1/0/2.0</td>
</tr>
<tr>
<td>qnn-vlan-300</td>
<td>b0:c6:9a:ca:3c:03</td>
<td>D</td>
<td>-</td>
<td>ge-1/0/2.0</td>
</tr>
</tbody>
</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)

### Purpose

Ensure that the MAC Address is learnt at the PE3 port (Branch Office B).

### Action

From operational mode, enter the **show ethernet-switching table** command.

```
user@host# show ethernet-switching table
```

**Verifying That the MAC Address is Learnt at the PE3 Port (Branch Office B)**

<table>
<thead>
<tr>
<th>Vlan name</th>
<th>MAC address</th>
<th>MAC flags</th>
<th>Age</th>
<th>Logical interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>qnn-vlan-300</td>
<td>b0:c6:9a:ca:3c:04</td>
<td>D</td>
<td>-</td>
<td>ge-1/0/2.0</td>
</tr>
</tbody>
</table>

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static)
**Action**  From operational mode, enter the `show ethernet-switching table` command.

```
user@host# show ethernet-switching table
```

<table>
<thead>
<tr>
<th>Vlan name</th>
<th>MAC address</th>
<th>MAC flags</th>
<th>Age</th>
<th>Logical interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>qng-vlan-300</td>
<td>b0:c6:9a:ca:3c:04</td>
<td>D</td>
<td></td>
<td>ge-1/0/2.0</td>
</tr>
</tbody>
</table>

MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static)

**Related Documentation**
CHAPTER 8

Configuration Statements for Layer 2 Bridge Domains

- action-priority on page 98
- bridge-domains on page 99
- bridge-options on page 101
- disable-action on page 102
- domain-type (Bridge Domains) on page 103
- enable-mac-move-action on page 104
- interface on page 105
- interface-mac-limit on page 106
- mac-statistics on page 108
- mac-table-size on page 110
- mac-table-aging-time on page 112
- no-irb-layer-2-copy on page 113
- no-mac-learning on page 114
- packet-action on page 117
- reopen-time on page 120
- routing-interface on page 121
- service-id on page 122
- static-mac on page 123
- vlan-id-list on page 125
- vlan-tags on page 126
### action-priority

<table>
<thead>
<tr>
<th><strong>Syntax</strong></th>
<th><code>action-priority number;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hierarchy Level</strong></td>
<td><code>[edit bridge-domains bridge-domain-name bridge-options interface interface-name]</code></td>
</tr>
<tr>
<td><strong>Release Information</strong></td>
<td>Statement introduced in Junos OS Release 13.2.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Configure the action priority value for an interface in a bridge domain on MAC move detection. This priority value is used to determine which interface should be blocked when a throttled MAC move is detected between two interfaces. The priority value can be between 0 and 7 inclusive. A higher value means lower priority. For example, if a MAC address move occurs between two interfaces with the action priority value set to 5 and 6, the interface with value 5 as the action priority value is blocked.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Required Privilege Level</strong></td>
<td>routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.</td>
</tr>
</tbody>
</table>
| **Related Documentation** | • Configuring a Bridge Domain on page 25  
• Configuring a Layer 2 Virtual Switch on page 48 |
bridge-domains

Syntax
bridge-domains {
  bridge-domain-name {
    bridge-options {
      ...bridge-options-configuration...
    }
    domain-type bridge;
    interface interface-name;
    no-irb-layer-2-copy;
    no-local-switching;
    routing-interface routing-interface-name;
    vlan-id (all | none | number);
    vlan-id-list [ vlan-id-numbers ];
    vlan-tags outer number inner number;
    bridge-options {
      interface interface-name {
        mac-pinning
        static-mac mac-address;
      }
      interface-mac-limit limit;
      mac-statistics;
      mac-table-size limit;
      no-mac-learning;
    }
  }
}

Hierarchy Level
[edit],
[edit logical-systems logical-system-name routing-instances routing-instance-name],
[edit routing-instances routing-instance-name]

Release Information
Statement introduced in Junos OS Release 8.4.
Support for logical systems added in Junos OS Release 9.6.
Support for the no-irb-layer-2-copy statement added in Junos OS Release 10.2.

Description
(MX Series routers only) Configure a domain that includes a set of logical ports that share the same flooding or broadcast characteristics in order to perform Layer 2 bridging.

On MX Series routers, Integrated routing and bridging (IRB) is not supported on Pseudowire Subscriber (PS) Logical Interface. Hence you cannot add IRB to bridge domain with PS interface, that is, you cannot configure IRB and PS interface in the same bridge domain. Note that adding IRB to a bridge-domain having PS interface causes kernel crash and continuous reboot of the router until the configuration is rolled back.

Options

- **bridge-domain-name** — Name of the bridge domain.
NOTE: You cannot use the slash (/) character as part of the bridge domain name. If you do, the configuration will not commit.

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege**
- routing—To view this statement in the configuration.
- routing-control—To add this statement to the configuration.

**Related Documentation**
- Configuring a Bridge Domain on page 25
- Configuring a Layer 2 Virtual Switch on page 48
**bridge-options**

**Syntax**

```plaintext
bridge-options {
  interface interface-name;
  static-mac static-mac-address;
}
global-mac-ip-limit limit;
interface-mac-ip-limit limit;
interface-mac-limit limit;
  packet-action drop;
}
mac-pinning
mac-statistics;
mac-ip-table-size limit;
mac-table-size limit;
mac-table-aging-time time;
nomac-learning;
}
```

**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]

**Release Information**

Statement introduced in Junos OS Release 8.4.
Support for logical systems added in Junos OS Release 9.6.
Statement (mac-pinning) introduced in Junos OS Release 16.2.

**Description**

(MX Series routers only) Configure Layer 2 learning and forwarding properties for a bridge domain 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**

- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
disable-action

Syntax  
disable-action;

Hierarchy Level  
[edit protocols l2-learning global-mac-move]

Release Information  
Statement introduced in Junos OS Release 13.2.

Description  
(MX Series routers only) Disable the MAC move action feature globally. MAC move detection configuration does exist, but the action is disabled.

Required Privilege  
Level  
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation  
• Configuring MAC Move Parameters
**domain-type (Bridge Domains)**

**Syntax**

```plaintext
domain-type bridge;
```

**ACX Series and MX Series**

```plaintext
[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**

```plaintext
[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.

**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](#)
- [Configuring a Bridge Domain on page 25](#)
### enable-mac-move-action

<table>
<thead>
<tr>
<th>Syntax</th>
<th>enable-mac-move-action;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy Level</td>
<td>[edit bridge-domains bridge-domain-name]</td>
</tr>
</tbody>
</table>

**Release Information**


**Description**

Enable the MAC move action feature at the bridge domain level. This statement blocks the logical interface for the bridge domain when a MAC move is detected 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 a Bridge Domain on page 25](#)
# 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 instance 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 on page 25
- Configuring a Layer 2 Virtual Switch on page 48
- Configuring a Layer 2 Virtual Switch on an EX Series Switch
- Tunnel Services Overview
- Tunnel Interface Configuration on MX Series Routers Overview
### interface-mac-limit

**Syntax**

```plaintext
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 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 on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
**mac-statistics**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>mac-statistics;</th>
</tr>
</thead>
</table>

**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 routinginstances routing-instance-name bridge-domains bridge-domain-name bridge-options],
- [edit logical-systems logical-system-name routinginstances routing-instance-name switch-options],
- [edit logical-systems logical-system-name switch-options],
- [edit routinginstances routing-instance-name bridge-domains bridge-domain-name bridge-options],
- [edit routinginstances routing-instance-name switch-options],
- [edit routinginstances 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.
Switches.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
Switches.
Support for EVPNs added in Junos OS Release 13.2 for MX 3D Series routers.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
Switches.
• Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83

• Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port

• Configuring EVPN Routing Instances

• Configuring EVPN Routing Instances on EX9200 Switches
**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 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 addresses

There 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.
- **Level**: routing-control—To add this statement to the configuration.

**Related Documentation**

- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
mac-table-aging-time

Syntax
mac-table-aging-time time;

Hierarchy Level
[edit logical-systems logical-system-name routing-instances routing-instance-name protocols vpls],
[edit routing-instances routing-instance-name protocols vpls]
[edit bridge-options],
[edit routing-instances routing-instance-name protocols evpn]

NOTE: For MX Series routers, the configuration statement is supported at the [bridge-options], [protocols vpls], and [protocols evpn] hierarchy levels only.

Release Information
Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 15.1 for MX Series routers.

Description
Modify the timeout interval for the MAC table.

For MX Series routers, you can use the global-mac-table-aging-time statement at the [edit protocols l2-learning] hierarchy level to configure the timeout interval at the global level or use the mac-table-aging-time to configure the timeout interval for a bridge domain or for a specific VPLS or EVPN instance. If multiple timeout interval values are configured on a router, the router determines the timeout interval value in the following order of priority:

- Timeout interval configured at the VPLS or EVPN instance
- Timeout interval configured for the bridge domain
- Global timeout interval configured on the router

NOTE: For MX Series routers, the timeout interval configuration feature is supported on routers with MPCs only.

Options
time—Specify the number of seconds to wait between MAC table clearings.
Range: 10 through 1,000,000 seconds
Default: 300 seconds

Required Privilege Level
routing—to view this statement in the configuration.
routing-control—to add this statement to the configuration.
Related Documentation

- Configuring the VPLS MAC Table Timeout Interval
- Configuring the MAC Table Timeout Interval on page 61

no-irb-layer-2-copy

Syntax

no-irb-layer-2-copy;

Hierarchy Level

[edit bridge-domains],
[edit logical-routers logical-router-name bridge-domains],
[edit routing-instances routing-instance-name bridge-domains]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

If you include this statement when using port mirroring with Integrated Routing and Bridging (IRB), then the packet is mirrored as a Layer 3 packet. By default, the packet is mirrored as a Layer 2 packet. This statement is also supported if a routing instance is set to type VPLS.

Usage Guidelines

See “Configuring a Bridge Domain” on page 25

Required Privilege

- view-level—To view this statement in the configuration.
- control-level—To add this statement to the configuration.

Related Documentation

- Configuring a Layer 2 Virtual Switch on page 48
### no-mac-learning

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| no-mac-learning; | 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]  
[edit vlans vlan-name]  
[edit vlans vlan-name switch-options] |
|-----------------------------|---------------------------------|

| EX Series and SRX Series Q-inQ Vlans | [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 bridge-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-domain-name bridge-options],  
[edit routing-instances routing-instance-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] |

ACX Series, MX Series, EX Series with ELS support, M Series, T Series |
**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 on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Understanding Bridging and VLANs on Switches
- Understanding Q-in-Q Tunneling and VLAN Translation
- Understanding Q-in-Q Tunneling and VLAN Translation
- Configuring Q-in-Q Tunneling on EX Series Switches
### 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 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-mac-limit limit],
[edit logical-systems logical-system-name switch-options interface interface-name 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 bridge-domains bridge-domain-name bridge-options 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-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 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 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]
```

**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 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 (URPF) 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

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>routing</td>
<td>To view this statement in the configuration.</td>
</tr>
<tr>
<td>routing-control</td>
<td>To add this statement to the configuration.</td>
</tr>
</tbody>
</table>

Related Documentation

- Configuring EVPN Routing Instances
- Configuring EVPN Routing Instances on EX9200 Switches
- Configuring MAC Limiting
- Configuring Persistent MAC Learning (ELS)
- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Overview
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
reopen-time

Syntax  reopen-time seconds;

Hierarchy Level  [edit protocols l2-learning global-mac-move]


Description  (MX Series routers only) Configure the value for the reopen timer.

Default  180 seconds

Options  seconds—Time duration after which the port is unblocked.

Required Privilege Level  routing—To view this statement in the configuration.
route-control—To add this statement to the configuration.

Related Documentation  • Configuring MAC Move Parameters
routing-interface

Syntax

```
routing-interface routing-interface-name;
```

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]
```

Release Information

Statement introduced in Junos OS Release 8.4.
Support for logical systems added in Junos OS Release 9.6.

Description

(MX Series routers only) Specify a routing interface to include in a bridge domain or a VPLS routing instance.

When you configure `routing-interface irb.x`, the VPLS connection comes up, even if no customer edge (CE) interfaces are configured. This works with one site configured, but not when multiple sites (multisite) are configured.

Options

```
routing-interface-name—Name of the routing interface to include in the bridge domain or the VPLS routing instance. The format of the routing interface name is irb.x, where x is the unit number of the routing interface you configured at the [edit interfaces irb] hierarchy level. For more information about how to configure a routing interface, see the Junos OS Network Interfaces Library for Routing Devices.
```

NOTE: You can specify only one routing interface for each bridge domain or VPLS instance.

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 on page 25
- Configuring a Layer 2 Virtual Switch on page 48
## service-id

<table>
<thead>
<tr>
<th>Syntax</th>
<th>service-id number;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy Level</td>
<td>[edit bridge-domains bridge-domain-name]</td>
</tr>
<tr>
<td>Release Information</td>
<td>Statement introduced in Junos OS Release 13.2</td>
</tr>
<tr>
<td>Description</td>
<td>Specify a service identifier to include in the packets sent to and from the multichassis link aggregation (MC-LAG) bridge domain when the VLAN identifier is set to none. This configuration facilitates media access control (MAC) and Address Resolution Protocol (ARP) synchronization among MC-LAG peers.</td>
</tr>
</tbody>
</table>

**NOTE:** The VLAN identifier none is supported only for IPv4 traffic.

| Options        | number—A valid service identifier. You must configure the same service identifier within the bridge domains of MC-LAG peers. |
|----------------|---------------------------------------------------------------------------------------------------------------------------------
| 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 on page 25  
  • Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances on page 38  
  • Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69  
  • bridge-domains on page 99 |
**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 EVPN added in Junos OS Release 13.2 for MX 3D Series routers. The `vlan-id` option is not available for EVPN.
[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 on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Adding a Static MAC Address Entry to the Ethernet Switching Table on a Switch with ELS Support
- Understanding VLANs on Security Devices
**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.
### vlan-tags

**Syntax**

```plaintext
tags outer number inner number;
```

**Hierarchy Level**

```plaintext
[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 on page 25
- Configuring a VLAN
- Configuring VLAN Identifiers for Bridge Domains and VPLS Routing Instances on page 38
- Configuring VLAN Identifiers for VLANs and VPLS Routing Instances
CHAPTER 9

Configuration Statements for Layer 2 Bridge Domains Functioning as Switches with Layer 2 Trunk Ports

- switch-options on page 128
- interface-mac-limit on page 129
- mac-statistics on page 131
- mac-table-size on page 133
- no-mac-learning on page 135
- packet-action on page 138
**switch-options**

**Syntax**

```
switch-options {
  interface interface-name {
    mac-pinning;
    interface-mac-limit limit;
  }
  interface-mac-limit limit {
    packet-action drop;
  }
  mac-statistics;
  mac-table-size limit {
    packet-action drop;
  }
  no-mac-learning;
  route-distinguisher (as-number:id | ip-address:id);
  service-id number; number;
  vrf-target {
    community;
    auto
    import community-name;
    export community-name;
  }
  vrf-import[ policy-names ];
  vrf-export[ policy-names ];
}
```

**Hierarchy Level**

```
[edit],
[edit logical-systems logical-system-name],
[edit logical-systems logical-system-name routing-instances routing-instance-name],
[edit routing-instances routing-instance-name]
```

**Release Information**

Statement introduced in Junos OS Release 9.2.
Support for logical systems added in Junos OS Release 9.6.
14.1x53-D10

**Description**

Configure Layer 2 learning and forwarding properties for a set of bridge domains.

**Options**

The remaining statements are explained separately. See CLI Explorer.

**Required Privilege Level**

routing—To view this statement in the configuration.
routeing-control—To add this statement to the configuration.

**Related Documentation**

- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
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 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.
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 on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
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-domain-name bridge-options],
[edit routing-instances routing-instance-name switch-options],
[edit routing-instances routing-instance-name protocols evpn],
[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 on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
• Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83

• Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port

• Configuring EVPN Routing Instances

• Configuring EVPN Routing Instances on EX9200 Switches
**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 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 addresses

There 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.
- **Level**: routing-control—To add this statement to the configuration.

**Related Documentation**

- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
**no-mac-learning**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-mac-learning;</td>
<td></td>
</tr>
</tbody>
</table>

**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-in-Q 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 bridge-domains bridge-domain-name bridge-options 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 EVPN 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 EVPN, 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 on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Understanding Bridging and VLANs on Switches
- Understanding Q-in-Q Tunneling and VLAN Translation
- Understanding Q-in-Q Tunneling and VLAN Translation
- Configuring Q-in-Q Tunneling on EX Series Switches
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 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-mac-limit (VPLS)],
[edit routing-instances routing-instance-name protocols evpn mac-table-size limit],
[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 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 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]
```

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 interface-name mac-table-size limit],
[edit vlans vlan-name switch-options interface interface-name interface-mac-limit limit],
[edit vlans vlan-name switch-options interface interface-name mac-table-size limit],


**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.

- **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.

**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.

**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
- Configuring Persistent MAC Learning (ELS)
- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Overview
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
CHAPTER 10

Configuration Statements for Layer 2 Address Learning and Forwarding

- l2-learning on page 142
- global-mac-limit on page 143
- global-mac-move on page 144
- global-mac-statistics on page 145
- global-mac-table-aging-time on page 146
- global-no-mac-learning on page 147
- interface-mac-limit on page 148
- notification-time on page 150
- packet-action on page 151
- threshold-count on page 154
- threshold-time on page 155
I2-learning

List of Syntax  
Syntax (MX Series, QFX Series, EX Series) on page 142  
Syntax (SRX Series) on page 142

Syntax (MX Series, QFX Series, EX Series)  
I2-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)  
I2-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-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 on page 18
- global-mac-table-aging-time on page 146
- global-mac-limit (Protocols)
- global-no-mac-learning on page 147
- global-mode (Protocols)

---

**global-mac-limit**

**Syntax**
```
global-mac-limit limit {
  packet-action drop;
}
```

**Hierarchy Level**

```
[edit protocols l2-learning]
```

**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.

**Description**
(MX Series routers and EX Series switches only) Limit the number of media access control (MAC) addresses learned from the logical interfaces on the router or switch.

**Default**
(MX Series) 393,215 MAC addresses
(EX9200) 524,287 MAC addresses

**Options**
- **limit**—Number of MAC addresses that can be learned systemwide.
  - **Range:** 20 through 1,048,575
  - 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**
- Limiting the Number of MAC Addresses Learned from Each Logical Interface on page 63
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.

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
- MAC Moves Loop Prevention in VPLS Network Overview
- Example: Configuring Loop Prevention in VPLS Network Due to MAC Moves
- virtual-mac
## global-mac-statistics

<table>
<thead>
<tr>
<th>Syntax</th>
<th><code>global-mac-statistics;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy Level</td>
<td>[edit protocols l2-learning]</td>
</tr>
<tr>
<td></td>
<td>Support for logical systems added in Junos OS Release 9.6.</td>
</tr>
<tr>
<td>Description</td>
<td>(MX Series routers and EX Series switches only) Enable MAC accounting for the entire router or switch.</td>
</tr>
<tr>
<td>Default</td>
<td>disabled</td>
</tr>
</tbody>
</table>
| Required Privilege Level | routing—To view this statement in the configuration.  
|                | routing-control—To add this statement to the configuration.     |
| Related        | Enabling MAC Accounting on page 62                             |

Copyright © 2018, Juniper Networks, Inc.
## 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 on page 61  
- Configuring MAC Table Aging on Switches  
- Example: Configuring VLANs on Security Devices
**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
- Understanding Q-in-Q Tunneling and VLAN Translation
- Example: Configuring VLANs on Security Devices
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 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 on page 69
• Layer 2 Learning and Forwarding for VLANs Overview
• Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
• Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
**notification-time**

**Syntax**  
nomination-time seconds;

**Hierarchy Level**  
[edit protocols l2-learning global-mac-move]

**Release Information**  
Statement introduced in Junos OS Release 9.4.  
Support for logical systems added in Junos OS Release 9.6.

**Description**  
(MX Series routers only) Configure the notification time value for MAC move reports that a MAC address moves before counting against the threshold values.

**Default**  
1 second

**Options**  
seconds—Time elapsed before MAC move reports are generated.

**Required Privilege Level**  
routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

**Related Documentation**  
- Configuring MAC Move Parameters
packet-action

Syntax

<table>
<thead>
<tr>
<th>Hierarchy Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>[edit bridge-domains bridge-domain-name bridge-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit bridge-domains bridge-domain-name bridge-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit bridge-domains bridge-domain-name bridge-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit logical-systems logical-system-name bridge-domains bridge-domain-name bridge-options interface-mac-limit limit],</td>
</tr>
<tr>
<td>[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],</td>
</tr>
<tr>
<td>[edit logical-systems logical-system-name routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit logical-systems logical-system-name routing-instances routing-instance-name switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit logical-systems logical-system-name routing-instances routing-instance-name switch-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit logical-systems logical-system-name switch-options interface interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit protocols l2-learning global-mac-limit limit],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name bridge-domains bridge-domain-name bridge-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name protocols evpn interface mac-limit (VPLS)],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name protocols evpn interface interface-name interface-mac-limit (VPLS)],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name protocols evpn mac-table-size limit],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit routing-instances routing-instance-name switch-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit switch-options interface mac-limit limit],</td>
</tr>
<tr>
<td>[edit switch-options mac-table-size limit],</td>
</tr>
<tr>
<td>[edit switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit vlans vlan-name switch-options interface interface-name interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit vlans vlan-name switch-options interface-mac-limit limit],</td>
</tr>
<tr>
<td>[edit vlans vlan-name switch-options mac-table-size limit]</td>
</tr>
</tbody>
</table>

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 interface-mac-limit limit]
- [edit vlans vlan-name 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 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 interface-name interface-mac-limit limit]
- [edit vlans vlan-name switch-options interface interface-name interface-mac-limit limit]


Support at the [edit switch-options interface interface-name interface-mac-limit limit]

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 QFX1002-60C 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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drop</td>
<td>Drop packets with new source MAC addresses, and do not learn the new source MAC addresses.</td>
</tr>
<tr>
<td>drop-and-log</td>
<td>(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.</td>
</tr>
<tr>
<td>log</td>
<td>(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.</td>
</tr>
<tr>
<td>none</td>
<td>(EX Series switches and QFX Series only) Forward packets with new source MAC addresses, and learn the new source MAC address.</td>
</tr>
<tr>
<td>shutdown</td>
<td>(EX Series switches and QFX Series only) Disable the specified interface, and generate an alarm, an SNMP trap, or a system log entry.</td>
</tr>
</tbody>
</table>

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.

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
- Configuring Persistent MAC Learning (ELS)
- Understanding Layer 2 Learning and Forwarding for Bridge Domains on page 69
- Layer 2 Learning and Forwarding for VLANs Overview
- Understanding Layer 2 Learning and Forwarding for Bridge Domains Functioning as Switches with Layer 2 Trunk Ports on page 83
- Layer 2 Learning and Forwarding for VLANs Overview
- Layer 2 Learning and Forwarding for VLANs Acting as a Switch for a Layer 2 Trunk Port
threshold-count

Syntax  
threshold-count count;

Hierarchy Level  
[edit protocols l2-learning global-mac-move]

Release Information  
Statement introduced in Junos OS Release 9.4.  
Support for logical systems added in Junos OS Release 9.6.

Description  
(MX Series routers only) Configure the threshold count value for MAC move reports.

Default  
50

Options  
count—Number of MAC moves needed in the notification time to generate a MAC move report.

Required Privilege  
Level  
routing—To view this statement in the configuration.  
routing-control—To add this statement to the configuration.

Related Documentation  
- Configuring MAC Move Parameters
threshold-time

Syntax

threshold-time seconds;

Hierarchy Level

[edit protocols l2-learning global-mac-move]

Release Information


Description

(MX Series routers only) Configure the threshold time value for MAC move reports when the MAC address moves at least a specified number of times (threshold count) in the configured interval.

Default

1 second

Options

seconds—Timer threshold before MAC move reports are generated.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

• Configuring MAC Move Parameters
CHAPTER 11

Operational Mode Commands for Layer 2 Bridge Domains

- clear bridge mac-table
- clear interfaces mac-database
- clear interfaces mac-database statistics
- show bridge domain
- show bridge flood
- show bridge mac-table
- show bridge statistics
- show interfaces queue
clear bridge mac-table

Syntax

```plaintext
clear bridge mac-table
  <bridge-domain (all | bridge-domain-name)>
  <instance instance-name>
  <interface interface-name>
  <learning-vlan id (all-vlan | learning-vlan-id)>
  <mac-address>
```

Release Information

Command introduced in Junos OS Release 8.4.

Description

(MX Series routers only) Clear learned Layer 2 address information from the media access control (MAC) address table.

Options

- **none**—Clear all learned Layer 2 address information from the MAC address table.
- **bridge-domain (all | bridge-domain-name)**—(Optional) Clear learned Layer 2 MAC addresses for all bridging domains or for the specified bridging domain.
- **instance instance-name**—(Optional) Clear learned Layer 2 MAC addresses for the specified routing instance.
- **interface interface-name**—(Optional) Clear learned Layer 2 MAC addresses for the specified interface.
- **learning-vlan-id (all-vlan | learning-vlan-id)**—(Optional) Clears learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.
- **mac-address**—(Optional) Clear the specified learned Layer 2 address from the MAC address table.

Required Privilege

**clear**

List of Sample Output

clear bridge mac-table on page 158

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear bridge mac-table

user@host> clear bridge mac-table
clear interfaces mac-database

Syntax  
clear interfaces mac-database (interface-name | aex) <mac-address mac-address>

Release Information  
Command introduced in Junos OS Release 8.3.  
Support for statement with the aex option introduced in Junos OS Release 15.1.

Description  
Clear learned media access control (MAC) addresses from the hardware and MAC database for Gigabit Ethernet IQ2 interfaces or aggregated Ethernet interfaces. Static MAC addresses configured by the operator are not cleared.

Options  
interface-name—Name of a physical or logical interface. When you clear a physical interface, all learned MAC addresses on all the logical interfaces under the physical interface are cleared.

aex—Name of aggregated Ethernet interface.

mac-address mac-address—(Optional) Clear only the specified MAC address.

Required Privilege Level  
view

List of Sample Output  
clear interfaces mac-database on page 159

Output Fields  
This command produces no output.

Sample Output

clear interfaces mac-database

user@host> clear interfaces mac-database ge-0/0/0.0
clear interfaces mac-database statistics

Syntax

```
clear interfaces mac-database statistics (interface-name | all)
```

Release Information

Command introduced in Junos OS Release 8.3.

Description

Clear statistics that are collected for every MAC address, including policer statistics, on a physical or logical interface or all interfaces.

Options

```
(interface-name | all)—Clear MAC database statistics for the specified physical or logical gigabit or 10-Gigabit Ethernet interface. Specify all to clear the MAC database statistics for all interfaces.
```

Required Privilege

view

List of Sample Output

clear interfaces mac-database statistics (Gigabit Ethernet) on page 160

Output Fields

This command produces no output.

Sample Output

clear interfaces mac-database statistics (Gigabit Ethernet)

```
user@host> clear interfaces mac-database statistics ge-0/1/0
```
show bridge domain

**Syntax**

```
show bridge domain
  <brief | detail | extensive>
  <bridge-domain (all | domain-name)>
  <instance instance-name>
  <operational>
```

**Release Information**

Command introduced in Junos OS Release 8.4.

**Description**

(MX Series routers only) Display bridge domain information.

**Options**

`none`—Display information for all bridge domains.

`brief | detail | extensive`—(Optional) Display the specified level of output.

`bridge-domain (all | domain-name)`—(Optional) Display information about all bridge domains or the specified bridge domain.

`instance instance-name`—(Optional) Display information for the specified routing instance.

`operational`—(Optional) Display information for the operational routing instances.

**Required Privilege Level**

view

**List of Sample Output**

`show bridge domain on page 161`
`show bridge domain brief on page 161`
`show bridge domain detail on page 162`

**Sample Output**

```
show bridge domain

<table>
<thead>
<tr>
<th>Instance</th>
<th>Bridging Domain</th>
<th>Type</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>vlan100</td>
<td>bridge</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bridge.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vs1</td>
<td>vlan200</td>
<td>bridge</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>bridge.0</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

show bridge domain brief

<table>
<thead>
<tr>
<th>Instance</th>
<th>Bridging Domain</th>
<th>Type</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>vlan100</td>
<td>bridge</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>bridge.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Copyright © 2018, Juniper Networks, Inc.
show bridge domain detail

user@host> show bridge domain detail

Routing Instance: vs1
  Bridging Domain: vlan100
  Router ID: 0.0.0.0
  Type: bridge  State: Active
  Interfaces:
    ge-11/0/3.0
    ge-11/1/4.100
    ge-11/1/1.100
    ge-11/1/0.100
    xe-10/2/0.100
    xe-10/0/0.100
  Tables:
    bridge.0 : 2 macs (2 active)

Routing Instance: vs1
  Bridging Domain: vlan200
  Router ID: 0.0.0.0
  Type: bridge  State: Active
  Interfaces:
    ge-11/1/0.200
    ge-11/1/1.200
    ge-11/1/4.200
    xe-10/0/0.200
    xe-10/2/0.200
  Tables:
    bridge.0 : 0 macs (0 active)
show bridge flood

Syntax

```
show bridge flood
  <brief | detail | extensive>
  <bridge-domain domain-name>
  <event-queue>
  <instance instance-name>
  <route (all-ce-flood | all ve-flood | alt-root-flood | bd-flood | mlp-flood | re-flood)>
```

Release Information

Command introduced in Junos OS Release 8.4.

Description

(MX Series routers only) Display bridging flooding information.

Options

none—Display all bridging flooding information for all bridging domains.

brief | detail | extensive—(Optional) Display the specified level of output.

bridge-domain domain-name—(Optional) Display bridging flooding information for the specified bridge domain.

event-queue—(Optional) Display the queue of pending bridge flood events.

instance instance-name—(Optional) Display bridging flooding information for the specified routing instance.

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 if no-local-switching is enabled.
- all-ve-flood—Display the route for flooding traffic to all VPLS edge routers 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 bridge domain 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.

Required Privilege Level

view

List of Sample Output

- show bridge flood on page 164
- show bridge flood brief on page 164
- show bridge flood detail on page 164
show bridge flood extensive on page 165

Output Fields to be provided

Sample Output

show bridge flood

user@host> show bridge flood

Name: __juniper_private1__
CEs: 0
VEs: 0
Flood Routes:
Prefix    Type          Owner                 NhType          NhIndex
0x36/16   MLP_FLOOD     __vs1+vlan100__       flood           426
0x3a/16   MLP_FLOOD     __vs1+vlan200__       flood           428

Name: vs1::vlan100
CEs: 6
VEs: 0
Flood Routes:
Prefix    Type          Owner                 NhType          NhIndex
0x35/16   ALL_FLOOD     __vs1+vlan100__       flood           425
0x35/16   RE_FLOOD      __vs1+vlan100__       flood           425
0x3780/17 ALT_ROOT_RT   ge-11/0/3.0          flood           425
0x3b80/17 ALT_ROOT_RT   ge-11/1/4.100        flood           425
0x3c80/17 ALT_ROOT_RT   ge-11/1/1.100        flood           425
0x3d80/17 ALT_ROOT_RT   ge-11/1/0.100        flood           425
0x3e80/17 ALT_ROOT_RT   xe-10/2/0.100        flood           425
0x3f80/17 ALT_ROOT_RT   xe-10/0/0.100        flood           425

Name: vs1::vlan200
CEs: 5
VEs: 0
Flood Routes:
Prefix    Type          Owner                 NhType          NhIndex
0x39/16   ALL_FLOOD     __vs1+vlan200__       flood           427
0x39/16   RE_FLOOD      __vs1+vlan200__       flood           427
0x4180/17 ALT_ROOT_RT   ge-11/1/0.200        flood           427
0x4080/17 ALT_ROOT_RT   ge-11/1/1.200        flood           427
0x4280/17 ALT_ROOT_RT   ge-11/1/4.200        flood           427
0x4480/17 ALT_ROOT_RT   xe-10/0/0.200        flood           427
0x4380/17 ALT_ROOT_RT   xe-10/2/0.200        flood           427

show bridge flood brief

user@host> show bridge flood brief

<table>
<thead>
<tr>
<th>Name</th>
<th>Active CEs</th>
<th>Active VEs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>juniper_private1</strong></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vs1::vlan100</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>vs1::vlan200</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

show bridge flood detail

user@host> show bridge flood detail

Name: __juniper_private1__
CEs: 0
VEs: 0
### Flood Routes:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Type</th>
<th>Owner</th>
<th>NhType</th>
<th>NhIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x36/16</td>
<td>MLP_FLOOD</td>
<td><strong>vs1+vlan100</strong></td>
<td>flood</td>
<td>426</td>
</tr>
<tr>
<td>0x3a/16</td>
<td>MLP_FLOOD</td>
<td><strong>vs1+vlan200</strong></td>
<td>flood</td>
<td>428</td>
</tr>
</tbody>
</table>

**Name:** vs1::vlan100

**CEs:** 6

**VEs:** 0

### Flood Routes:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Type</th>
<th>Owner</th>
<th>NhType</th>
<th>NhIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x35/16</td>
<td>ALL_FLOOD</td>
<td><strong>vs1+vlan100</strong></td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x35/16</td>
<td>RE_FLOOD</td>
<td><strong>vs1+vlan100</strong></td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x3780/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/0/3.0</td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x3b80/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/1/4.100</td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x3c80/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/1/1.100</td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x3d80/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/1/0.100</td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x3e80/17</td>
<td>ALT_ROOT_RT</td>
<td>xe-10/2/0.100</td>
<td>flood</td>
<td>425</td>
</tr>
<tr>
<td>0x3f80/17</td>
<td>ALT_ROOT_RT</td>
<td>xe-10/0/0.100</td>
<td>flood</td>
<td>425</td>
</tr>
</tbody>
</table>

**Name:** vs1::vlan200

**CEs:** 5

**VEs:** 0

### Flood Routes:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Type</th>
<th>Owner</th>
<th>NhType</th>
<th>NhIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x39/16</td>
<td>ALL_FLOOD</td>
<td><strong>vs1+vlan200</strong></td>
<td>flood</td>
<td>427</td>
</tr>
<tr>
<td>0x39/16</td>
<td>RE_FLOOD</td>
<td><strong>vs1+vlan200</strong></td>
<td>flood</td>
<td>427</td>
</tr>
<tr>
<td>0x4180/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/1/0.200</td>
<td>flood</td>
<td>427</td>
</tr>
<tr>
<td>0x4080/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/1/1.200</td>
<td>flood</td>
<td>427</td>
</tr>
<tr>
<td>0x4280/17</td>
<td>ALT_ROOT_RT</td>
<td>ge-11/1/4.200</td>
<td>flood</td>
<td>427</td>
</tr>
<tr>
<td>0x4480/17</td>
<td>ALT_ROOT_RT</td>
<td>xe-10/0/0.200</td>
<td>flood</td>
<td>427</td>
</tr>
<tr>
<td>0x4380/17</td>
<td>ALT_ROOT_RT</td>
<td>xe-10/2/0.200</td>
<td>flood</td>
<td>427</td>
</tr>
</tbody>
</table>

---

**show bridge flood extensive**

```
user@host> show bridge flood extensive

Name: __juniper_private1__

**CEs:** 0

**VEs:** 0

Flood route prefix: 0x36/16
Flood route type: MLP_FLOOD
Flood route owner: __vs1+vlan100__
Nexthop type: flood
Nexthop index: 426

Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>lc-11/0/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lc-10/2/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lc-10/0/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lc-11/1/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x3a/16
Flood route type: MLP_FLOOD
Flood route owner: __vs1+vlan200__
Nexthop type: flood
Nexthop index: 428

Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>lc-10/0/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lc-10/2/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lc-11/1/0.32769</td>
<td>LC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name: vs1::vlan100
```
<table>
<thead>
<tr>
<th>Flood route prefix: 0x35/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flood route type: ALL_FLOOD</td>
</tr>
<tr>
<td>Flood route owner: <strong>vs1+vlan100</strong></td>
</tr>
<tr>
<td>Nexthop type: flood</td>
</tr>
<tr>
<td>Nexthop index: 425</td>
</tr>
<tr>
<td>Interfaces Flooding to:</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>ge-11/0/3.0</td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flood route prefix: 0x3780/17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flood route type: ALT_ROOT_RT</td>
</tr>
<tr>
<td>Flood route owner: ge-11/0/3.0</td>
</tr>
<tr>
<td>Nexthop type: flood</td>
</tr>
<tr>
<td>Nexthop index: 425</td>
</tr>
<tr>
<td>Interfaces Flooding to:</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>ge-11/0/3.0</td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flood route prefix: 0x3b80/17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flood route type: ALT_ROOT_RT</td>
</tr>
<tr>
<td>Flood route owner: ge-11/1/4.100</td>
</tr>
<tr>
<td>Nexthop type: flood</td>
</tr>
<tr>
<td>Nexthop index: 425</td>
</tr>
<tr>
<td>Interfaces Flooding to:</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>ge-11/0/3.0</td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flood route prefix: 0x3c80/17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flood route type: ALT_ROOT_RT</td>
</tr>
<tr>
<td>Flood route owner: ge-11/1/4.100</td>
</tr>
<tr>
<td>Nexthop type: flood</td>
</tr>
<tr>
<td>Nexthop index: 425</td>
</tr>
<tr>
<td>Interfaces Flooding to:</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>ge-11/0/3.0</td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
</tr>
</tbody>
</table>
Flood route type: ALT_ROOT_RT
Flood route owner: ge-11/1/1.100
Nexthop type: flood
Nexthop index: 425
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/0/3.0</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x3d80/17
Flood route type: ALT_ROOT_RT
Flood route owner: ge-11/1/0.100
Nexthop type: flood
Nexthop index: 425
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/0/3.0</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x3e80/17
Flood route type: ALT_ROOT_RT
Flood route owner: xe-10/2/0.100
Nexthop type: flood
Nexthop index: 425
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/0/3.0</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x3f80/17
Flood route type: ALT_ROOT_RT
Flood route owner: xe-10/0/0.100
Nexthop type: flood
Nexthop index: 425
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/0/3.0</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Name: vs1::vlan200
CEs: 5
VES: 0

Flood route prefix: 0x39/16
Flood route type: ALL_FLOOD
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x39/16
Flood route type: RE_FLOOD
Flood route owner: __vs1+vlan200__
Nexthop type: flood
Nexthop index: 427
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x4180/17
Flood route type: ALT_ROOT_RT
Flood route owner: ge-11/1/0.200
Nexthop type: flood
Nexthop index: 427
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x4080/17
Flood route type: ALT_ROOT_RT
Flood route owner: ge-11/1/1.200
Nexthop type: flood
Nexthop index: 427
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x4280/17
Flood route type: ALT_ROOT_RT
Flood route owner: ge-11/1/4.200
Nexthop type: flood
Nexthop index: 427
Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
xe-10/0/0.200 CE
xe-10/2/0.200 CE

Flood route prefix: 0x4480/17
Flood route type: ALT_ROOT_RT
Flood route owner: xe-10/0/0.200
Nexthop type: flood
Nexthop index: 427

Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Flood route prefix: 0x4380/17
Flood route type: ALT_ROOT_RT
Flood route owner: xe-10/2/0.200
Nexthop type: flood
Nexthop index: 427

Interfaces Flooding to:
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>NhType</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>CE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**show bridge mac-table**

**Syntax**

```
show bridge mac-table

<age>
<brief | count | detail | extensive>
<brIDGE-domain (all | bridge-domain-name)>
<global-count>
<instance instance-name>
<interface interface-name>
<mac-address>
<instance instance-name>
<vlan-id (all-vlan | vlan-id)>
```

**Release Information**

- Command introduced in Junos OS Release 8.4.
- Command introduced in Junos OS Release 15.1
- Support for PBB-EVPN instance added in Junos OS Release 16.1
- MAC Flag P to indicate a MAC Pinned interface introduced in Junos OS 16.2

**Description**

(MX Series routers only) Display Layer 2 MAC address information.

**Options**

- **none**—Display all learned Layer 2 MAC address information.
- **age**—(Optional) Display age of a single mac-address.
- **brief | count | detail | extensive**—(Optional) Display the specified level of output.
- **bridge-domain (all | bridge-domain-name)**—(Optional) Display learned Layer 2 MAC addresses for all bridging domains or for the specified bridging domain.
- **global-count**—(Optional) Display the total number of learned Layer 2 MAC addresses on the system.
- **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.
- **mac-address**—(Optional) Display the specified learned Layer 2 MAC address information.
- **vlan-id (all-vlan | vlan-id)**—(Optional) Display learned Layer 2 MAC addresses for all VLANs or for the specified VLAN.

**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**

<table>
<thead>
<tr>
<th>List of Sample Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>show bridge mac-table on page 172</td>
</tr>
<tr>
<td>show bridge mac-table (with Layer 2 Services over GRE Interfaces) on page 172</td>
</tr>
<tr>
<td>show bridge mac-table (with VXLAN enabled) on page 173</td>
</tr>
<tr>
<td>show bridge mac-table age (for GE interface) on page 173</td>
</tr>
<tr>
<td>show bridge mac-table age (for AE interface) on page 173</td>
</tr>
<tr>
<td>show bridge mac-table count on page 173</td>
</tr>
<tr>
<td>show bridge mac-table detail on page 174</td>
</tr>
<tr>
<td>show bridge mac-table instance pbb-evpn on page 174</td>
</tr>
<tr>
<td>show bridge mac-table on page 174</td>
</tr>
</tbody>
</table>

**Output Fields**

Table 7 on page 171 describes the output fields for the `show bridge mac-table` command. Output fields are listed in the approximate order in which they appear.

**Table 7: show bridge mac-table Output Fields**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>Age of a single mac-address.</td>
</tr>
<tr>
<td>Routing instance</td>
<td>Name of the routing instance.</td>
</tr>
<tr>
<td>Bridging domain</td>
<td>Name of the bridging domain.</td>
</tr>
<tr>
<td>MAC address</td>
<td>MAC address or addresses learned on a logical interface.</td>
</tr>
<tr>
<td>MAC flags</td>
<td>Status of MAC address learning properties for each interface:</td>
</tr>
<tr>
<td></td>
<td>• S—Static MAC address is configured.</td>
</tr>
<tr>
<td></td>
<td>• D—Dynamic MAC address is configured.</td>
</tr>
<tr>
<td></td>
<td>• L—Locally learned MAC address is configured.</td>
</tr>
<tr>
<td></td>
<td>• C—Control MAC address is configured.</td>
</tr>
<tr>
<td></td>
<td>• SE—MAC accounting is enabled.</td>
</tr>
<tr>
<td></td>
<td>• NM—Non-configured MAC.</td>
</tr>
<tr>
<td></td>
<td>• R—Remote PE MAC address is configured.</td>
</tr>
<tr>
<td></td>
<td>• P—MAC Pinned interface is configured.</td>
</tr>
<tr>
<td>Logical interface</td>
<td>Name of the logical interface.</td>
</tr>
<tr>
<td>MAC count</td>
<td>Number of MAC addresses learned on the specific routing instance or interface.</td>
</tr>
<tr>
<td>Learning interface</td>
<td>Name of the logical interface on which the MAC address was learned.</td>
</tr>
<tr>
<td>Learning VLAN</td>
<td>VLAN ID of the routing instance or bridge domain in which the MAC address was learned.</td>
</tr>
<tr>
<td>VXLAN ID/VXLAN</td>
<td>VXLAN Network Identifier (VNI).</td>
</tr>
</tbody>
</table>
Table 7: show bridge mac-table Output Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 2 flags</td>
<td>Debugging flags signifying that the MAC address is present in various lists.</td>
</tr>
<tr>
<td>Epoch</td>
<td>Spanning Tree Protocol epoch number identifying when the MAC address was learned.</td>
</tr>
<tr>
<td>Sequence number</td>
<td>Sequence number assigned to this MAC address. Used for debugging.</td>
</tr>
<tr>
<td>Learning mask</td>
<td>Mask of the Packet Forwarding Engines where this MAC address was learned. Used for debugging.</td>
</tr>
<tr>
<td>IPC generation</td>
<td>Creation time of the logical interface when this MAC address was learned. Used for debugging.</td>
</tr>
</tbody>
</table>

Sample Output

show bridge mac-table

user@host> show bridge mac-table

MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC)

Routing instance : default-switch
Bridging domain : test1, VLAN : 1

<table>
<thead>
<tr>
<th>MAC address</th>
<th>MAC flags</th>
<th>Logical interface</th>
<th>NH</th>
<th>RTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>01:00:0c:cc:cc:cc</td>
<td>S,NM</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01:00:0c:cc:cc:cd</td>
<td>S,NM</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01:00:0c:cd:cd:d0</td>
<td>S,NM</td>
<td>NULL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64:87:88:6a:17:d0</td>
<td>D</td>
<td>ae0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64:87:88:6a:17:f0</td>
<td>D</td>
<td>ae0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

show bridge mac-table (with Layer 2 Services over GRE Interfaces)

user@host> show bridge mac-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
Bridging domain : vlan-1, VLAN : 1

<table>
<thead>
<tr>
<th>MAC address</th>
<th>MAC flags</th>
<th>Logical interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:01:01:00:01:f7</td>
<td>D,SE</td>
<td>gr-1/2/10.0</td>
</tr>
<tr>
<td>00:03:00:32:01:f7</td>
<td>D,SE</td>
<td>gr-1/2/10.0</td>
</tr>
<tr>
<td>00:00:21:11:11:10</td>
<td>DL</td>
<td>ge-1/0/0.0</td>
</tr>
<tr>
<td>00:00:21:11:11:11</td>
<td>DL</td>
<td>ge-1/1/0.0</td>
</tr>
</tbody>
</table>

Routing instance : default-switch
Bridging domain : vlan-2, VLAN : 2

<table>
<thead>
<tr>
<th>MAC</th>
<th>Logical</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
show bridge mac-table (with VXLAN enabled)

```
user@host> show bridge mac-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
Bridging domain : vlan-1, VLAN : 1
VXLAN: Id : 100, Multicast group: 233.252.0.1
MAC address flags interface
00:01:01:00:01:f7  D,SE     vtep.1052010
00:03:00:32:01:f7  D,SE     vtep.1052011
00:00:21:11:11:10 DL       ge-1/0/0.0
00:00:21:11:11:11 DL       ge-1/0/0.0

Routing instance : default-switch
Bridging domain : vlan-2, VLAN : 2, VXLAN : 200
VXLAN: Id : 200, Multicast group: 233.252.0.2
MAC address flags interface
00:02:01:33:01:f7  D,SE     vtep.1052010
00:04:00:14:01:f7  D,SE     vtep.1052011
00:00:21:11:21:10 DL       ge-1/0/0.1
00:00:21:11:21:11 DL       ge-1/0/0.1
```

show bridge mac-table age (for GE interface)

```
user@host> show vpls mac-table age 00:02:03:aa:bb:1a instance vpls_instance_1
MAC Entry Age information
Current Age: 4 seconds
```

show bridge mac-table age (for AE interface)

```
user@host> show vpls mac-table age 00:02:03:aa:bb:1a instance vpls_instance_1
MAC Entry Age information
Current Age on FPC1: 102 seconds
Current Age on FPC2: 94 seconds
```

show bridge mac-table count

```
user@host> show bridge mac-table count
2 MAC address learned in routing instance vs1 bridge domain vlan100

MAC address count per interface within routing instance:
Logical interface     MAC count
ge-11/0/3.0           1
ge-11/1/4.100          0
```
show bridge mac-table detail

user@host> show bridge mac-table detail
MAC address: 00:00:00:19:1c:db
Routing instance: vs1
Bridging domain: vlan100
Learning interface: ge-11/0/3.0 Learning VLAN: 0
Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
Epoch: 4 Sequence number: 0
Learning mask: 0x800 IPC generation: 0

MAC address: 00:00:00:59:3a:2f
Routing instance: vs1
Bridging domain: vlan100
Learning interface: xe-10/2/0.100 Learning VLAN: 0
Layer 2 flags: in_ifd, in_ifl, in_vlan, kernel
Epoch: 7 Sequence number: 0
Learning mask: 0x400 IPC generation: 0

show bridge mac-table instance pbb-evpn

user@host> show bridge mac-table instance pbb-evpn
Routing instance: pbb-evpn
Bridging domain : isid-bd10000, ISID : 10000

<table>
<thead>
<tr>
<th>MAC address</th>
<th>MAC flags</th>
<th>Logical flags</th>
<th>Logical interface</th>
<th>NH</th>
<th>NH Index</th>
<th>RTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:19:e2:b0:76:eb</td>
<td>D</td>
<td>cbp.1000</td>
<td></td>
<td>1048576</td>
<td></td>
<td></td>
</tr>
<tr>
<td>aa:bb:cc:dd:ee:f2</td>
<td>DC</td>
<td></td>
<td></td>
<td>1048576</td>
<td>1048576</td>
<td></td>
</tr>
<tr>
<td>aa:bb:cc:dd:ee:f3</td>
<td>DC</td>
<td></td>
<td></td>
<td>1048575</td>
<td>1048575</td>
<td></td>
</tr>
</tbody>
</table>

show bridge mac-table

user@host> run show bridge mac-table
### MAC flags (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC, O -OVSDB MAC, SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC, P -Pinned MAC)

Routing instance : VS-541  
Bridging domain : 541, VLAN : 541  
MAC MAC Logical NH RTR  
address flags interface Index ID  
00:00:01:00:00:01 D P RC xe-0/0/3.0  
00:00:02:00:00:01 D P xe-0/0/3.0

- **S** -static MAC  
- **D** -dynamic MAC  
- **L** -locally learned  
- **C** -Control MAC  
- **O** -OVSDB MAC  
- **SE** -Statistics enabled  
- **NM** -Non configured MAC  
- **R** -Remote PE MAC  
- **P** -Pinned MAC
**show bridge statistics**

**Syntax**

```
show bridge statistics
<bridge-domain domain-name>
<instance instance-name>
```

**Release Information**

Command introduced in Junos OS Release 8.4.

**Description**

(MX Series routers only) Display bridge statistics.

**Options**

- `none`—Display bridge statistics for all bridge domains in all routing instances.
- `bridge-domain domain-name`—(Optional) Display statistics for the specified bridge domain.
- `instance instance-name`—(Optional) Display statistics for the specified routing instance.

**Required Privilege**

view

**List of Sample Output**

show bridge statistics on page 176

**Sample Output**

```
show bridge statistics

user@host> show bridge statistics

Information for routing instance:

Routing instance : __juniper_private1__
  Index: 1                        Sequence number: 0
  MAC limit: 5000                 MACs learned: 0
  Static MACs learned: 0          Non config Static MACs learned: 0
  Handle: 0x829e800

Information for routing instance:

Routing instance : vs1
  Bridging domain : vlan100
  Index: 3                        Sequence number: 0
  MAC limit: 5120                 MACs learned: 2
  Static MACs learned: 0          Non config Static MACs learned: 0
  Handle: 0x829e400
  Flags: Bridge instance, Config defined, VLAN : 100
  Local interface: ge-11/0/3.0, Index: 79
    Broadcast packets:                     1
    Broadcast bytes :                    65
    Multicast packets:                   0
    Multicast bytes :                    0
    Flooded packets :                     0
    Flooded bytes :                      0
    Unicast packets :             358624489
```
<table>
<thead>
<tr>
<th>Interface</th>
<th>Unicast packets</th>
<th>Unicast bytes</th>
<th>Broadcast packets</th>
<th>Broadcast bytes</th>
<th>Multicast packets</th>
<th>Multicast bytes</th>
<th>Flooded packets</th>
<th>Flooded bytes</th>
<th>Unicast packets</th>
<th>Unicast bytes</th>
<th>Current MAC count</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/1/4.100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (Limit 1024)</td>
<td></td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (Limit 1024)</td>
<td></td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (Limit 1024)</td>
<td></td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
<td>358627393</td>
<td>23310781065</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (Limit 1024)</td>
<td></td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 (Limit 1024)</td>
<td></td>
</tr>
</tbody>
</table>

Information for routing instance:

Routing instance: vs1
Bridging domain: vlan200
Index: 4
MAC limit: 5120
Static MACs learned: 0
Non config Static MACs learned: 0
### Handle: 0x829e600
- **Flags:** Bridge instance, Config defined, VLAN : 200
- **Local interface:** ge-11/1/0.200, Index: 90
  - 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-11/1/1.200, Index: 91
- 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-11/1/4.200, Index: 92
- 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: xe-10/0/0.200, Index: 93
- Broadcast packets: 4
- Broadcast bytes: 260
- 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: xe-10/2/0.200, Index: 94
- 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)
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 8 on page 180. 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 8 on page 180.

**Table 8: Layer 2 Overhead and Transmitted Packets or Byte Counts**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Fragmentation</th>
<th>LFI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First fragmentation</td>
<td>Second to n fragmentations</td>
</tr>
<tr>
<td></td>
<td>Bytes</td>
<td>Bytes</td>
</tr>
<tr>
<td>MLPPP (Long)</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>MLPPP (short)</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>MLFR (FRF15)</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>MFR (FRF16)</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>MCMLPPP (Long)</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>MCMLPPP (Short)</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

**Layer 2 Statistics—Fragmentation Overhead Calculation**

**MLPPP/MC-MLPPP Overhead details:**

<table>
<thead>
<tr>
<th>Fragment 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outer PPP header : 4 bytes</td>
</tr>
<tr>
<td>Long or short sequence MLPPP header : 4 bytes or 2 bytes</td>
</tr>
<tr>
<td>Inner PPP header : 1 byte</td>
</tr>
<tr>
<td>HDLC flag and FCS bytes : 4 bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragments 2 .. n:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outer PPP header : 4 bytes</td>
</tr>
<tr>
<td>Long or short sequence MLPPP header : 4 bytes or 2 bytes</td>
</tr>
<tr>
<td>HDLC flag and FCS bytes : 4 bytes</td>
</tr>
</tbody>
</table>

**MLFR (FRF15) Overhead details:**

<table>
<thead>
<tr>
<th>Fragment 1:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framerelay header : 2 bytes</td>
</tr>
<tr>
<td>Control, NLPID : 2 bytes</td>
</tr>
<tr>
<td>Fragmentation header : 2 bytes</td>
</tr>
<tr>
<td>Inner proto : 2 bytes</td>
</tr>
<tr>
<td>HDLC flag and FCS : 4 bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fragments 2 ... n:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framerelay header : 2 bytes</td>
</tr>
</tbody>
</table>
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 250 byte. 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 Queueing 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 188
- show interfaces queue (Aggregated Ethernet on a T320 Router) on page 189
- show interfaces queue (Gigabit Ethernet on a T640 Router) on page 191
- show interfaces queue aggregate (Gigabit Ethernet Enhanced DPC) on page 192
- show interfaces queue (Gigabit Ethernet IQ2 PIC) on page 195
- show interfaces queue both-ingress-egress (Gigabit Ethernet IQ2 PIC) on page 198
Output Fields

Table 9 on page 183 lists the output fields for the `show interfaces queue` command. Output fields are listed in the approximate order in which they appear.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical interface</td>
<td>Name of the physical interface.</td>
</tr>
<tr>
<td>Enabled</td>
<td>State of the interface. Possible values are described in the “Enabled Field” section under Common Output Fields Description.</td>
</tr>
<tr>
<td>Interface index</td>
<td>Physical interface's index number, which reflects its initialization sequence.</td>
</tr>
<tr>
<td>SNMP ifIndex</td>
<td>SNMP index number for the interface.</td>
</tr>
<tr>
<td>Forwarding classes supported</td>
<td>Total number of forwarding classes supported on the specified interface.</td>
</tr>
<tr>
<td>Forwarding classes in use</td>
<td>Total number of forwarding classes in use on the specified interface.</td>
</tr>
<tr>
<td>Ingress queues supported</td>
<td>On Gigabit Ethernet IQ2 PICs only, total number of ingress queues supported on the specified interface.</td>
</tr>
<tr>
<td>Ingress queues in use</td>
<td>On Gigabit Ethernet IQ2 PICs only, total number of ingress queues in use on the specified interface.</td>
</tr>
<tr>
<td>Output queues supported</td>
<td>Total number of output queues supported on the specified interface.</td>
</tr>
<tr>
<td>Output queues in use</td>
<td>Total number of output queues in use on the specified interface.</td>
</tr>
<tr>
<td>Egress queues supported</td>
<td>Total number of egress queues supported on the specified interface.</td>
</tr>
<tr>
<td>Egress queues in use</td>
<td>Total number of egress queues in use on the specified interface.</td>
</tr>
</tbody>
</table>
Table 9: show interfaces queue Output Fields (continued)

<table>
<thead>
<tr>
<th>Field Name (Ingress)</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue counters</td>
<td>CoS queue number and its associated user-configured forwarding class name. Displayed on IQ2 interfaces.</td>
</tr>
<tr>
<td>Queued packets</td>
<td>Number of queued packets.</td>
</tr>
<tr>
<td><strong>NOTE:</strong> This field is not supported on QFX5100, QFX5110, QFX5200, and QFX5210 switches due to hardware limitations.</td>
<td></td>
</tr>
<tr>
<td>Transmitted packets</td>
<td>Number of transmitted packets.</td>
</tr>
<tr>
<td>Dropped packets</td>
<td>Number of packets dropped by the ASIC’s RED mechanism.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td>Queue number.</td>
</tr>
<tr>
<td>Forwarding classes</td>
<td>Forwarding class name.</td>
</tr>
<tr>
<td>Queued Packets</td>
<td>Number of packets queued to this queue.</td>
</tr>
<tr>
<td><strong>NOTE:</strong> 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 &quot;Additional Information&quot; on page 181.</td>
<td></td>
</tr>
<tr>
<td>Queued Bytes</td>
<td>Number of bytes queued to this queue. The byte counts vary by interface hardware. For more information, see Table 10 on page 187. 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 &quot;Additional Information&quot; on page 181.</td>
</tr>
<tr>
<td><strong>NOTE:</strong> This field is not supported on QFX5100, QFX5110, QFX5200, and QFX5210 switches due to hardware limitations.</td>
<td></td>
</tr>
<tr>
<td>Transmitted Packets</td>
<td>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.</td>
</tr>
<tr>
<td><strong>NOTE:</strong> For Layer 2 statistics, see &quot;Overhead for Layer 2 Statistics&quot; on page 179</td>
<td></td>
</tr>
</tbody>
</table>
Table 9: show interfaces queue Output Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted Bytes</td>
<td>Number of bytes transmitted by this queue. The byte counts vary by interface hardware. For more information, see Table 10 on page 187.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> For Layer 2 statistics, see “Overhead for Layer 2 Statistics” on page 179</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>Number of packets dropped because of tail drop.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> Starting with Junos OS 18.3R1, the Tail-dropped packets counter is supported on PTX Series Packet Transport Routers.</td>
</tr>
<tr>
<td>RL-dropped packets</td>
<td>Number of packets dropped due to rate limiting.</td>
</tr>
<tr>
<td></td>
<td>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 181.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> The RL-dropped packets counter is not supported on the PTX Series Packet Transport Routers, and is omitted from the output.</td>
</tr>
<tr>
<td>RL-dropped bytes</td>
<td>Number of bytes dropped due to rate limiting.</td>
</tr>
<tr>
<td></td>
<td>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 181.</td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>Number of packets dropped because of random early detection (RED).</td>
</tr>
<tr>
<td></td>
<td>• (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:</td>
</tr>
<tr>
<td></td>
<td>• Low, non-TCP—Number of low-loss priority non-TCP packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• Low, TCP—Number of low-loss priority TCP packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• High, non-TCP—Number of high-loss priority non-TCP packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• High, TCP—Number of high-loss priority TCP packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• (MX Series routers with enhanced DPCs, and T Series routers with enhanced FPCs only) The output classifies dropped packets into the following categories:</td>
</tr>
<tr>
<td></td>
<td>• Low—Number of low-loss priority packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• Medium-low—Number of medium-low loss priority packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• Medium-high—Number of medium-high loss priority packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>• High—Number of high-loss priority packets dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> 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.</td>
</tr>
</tbody>
</table>
### Table 9: show interfaces queue Output Fields (continued)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED-dropped bytes</td>
<td>Number of bytes dropped because of RED. The byte counts vary by interface hardware. For more information, see Table 10 on page 187.</td>
</tr>
<tr>
<td></td>
<td>- (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:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Low, non-TCP</strong>—Number of low-loss priority non-TCP bytes dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Low, TCP</strong>—Number of low-loss priority TCP bytes dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>- <strong>High, non-TCP</strong>—Number of high-loss priority non-TCP bytes dropped because of RED.</td>
</tr>
<tr>
<td></td>
<td>- <strong>High, TCP</strong>—Number of high-loss priority TCP bytes dropped because of RED.</td>
</tr>
<tr>
<td>NOTE:</td>
<td>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.</td>
</tr>
<tr>
<td>Queue-depth bytes</td>
<td>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.</td>
</tr>
<tr>
<td>Queue-depth bytes</td>
<td>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.</td>
</tr>
<tr>
<td>Last-packet enqueued</td>
<td>Starting with Junos OS Release 16.1, <strong>Last-packet enqueued</strong> output field is introduced. If <strong>packet-timestamp</strong> is enabled for an FPC, shows the day, date, time, and year in the format <code>day-of-the-week month day-date hh:mm:ss yyyy</code> 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.</td>
</tr>
</tbody>
</table>

Byte counts vary by interface hardware. Table 10 on page 187 shows how the byte counts on the outbound interfaces vary depending on the interface hardware. Table 10 on page 187 is based on the assumption that outbound interfaces are sending IP traffic with 478 bytes per packet.
### Table 10: Byte Count by Interface Hardware

<table>
<thead>
<tr>
<th>Interface Hardware</th>
<th>Output Level</th>
<th>Byte Count Includes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gigabit Ethernet IQ and IQE PICs</td>
<td>Interface</td>
<td>Queued: 490 bytes per packet, representing 478 bytes of Layer 3 packet + 12 bytes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmitted: 490 bytes per packet, representing 478 bytes of Layer 3 packet + 12 bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RED dropped: 496 bytes per packet representing 478 bytes of Layer 3 packet + 18 bytes</td>
<td></td>
</tr>
<tr>
<td>Packet forwarding component</td>
<td></td>
<td>Queued: 478 bytes per packet, representing 478 bytes of Layer 3 packet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmitted: 478 bytes per packet, representing 478 bytes of Layer 3 packet</td>
<td></td>
</tr>
<tr>
<td>Non-IQ PIC</td>
<td>Interface</td>
<td>T Series, TX Series, T1600, and MX Series routers:</td>
<td>The Layer 2 overhead is 14 bytes for non-VLAN traffic and 18 bytes for VLAN traffic.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Queued: 478 bytes of Layer 3 packet.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Transmitted: 478 bytes of Layer 3 packet.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>T4000 routers with Type 5 FPCs:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>M Series routers:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Queued: 478 bytes of Layer 3 packet.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Transmitted: 478 bytes of Layer 3 packet + the full Layer 2 overhead.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PTX Series Packet Transport Routers:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Queued: The sum of the transmitted bytes and the RED dropped bytes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Transmitted: Full Layer 2 overhead (including all L2 encapsulation and CRC) + 12 Inter-packet gap + 8 for the preamble.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 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).</td>
<td></td>
</tr>
</tbody>
</table>
Table 10: Byte Count by Interface Hardware (continued)

<table>
<thead>
<tr>
<th>Interface Hardware</th>
<th>Output Level</th>
<th>Byte Count Includes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQ and IQE PICs with a SONET/SDH interface</td>
<td>Interface</td>
<td>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</td>
<td>The additional 4 bytes are for the Layer 2 Point-to-Point Protocol (PPP) header.</td>
</tr>
<tr>
<td></td>
<td>Packet forwarding component</td>
<td>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</td>
<td>For transmitted packets, the additional 8 bytes includes 4 bytes for the PPP header and 4 bytes for a cookie.</td>
</tr>
<tr>
<td>Non-IQ PIC with a SONET/SDH interface</td>
<td>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</td>
<td>For transmitted packets, the additional 5 bytes includes 4 bytes for the PPP header and 1 byte for the packet loss priority (PLP).</td>
<td></td>
</tr>
<tr>
<td>Interfaces configured with Frame Relay Encapsulation</td>
<td>Interface</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>1-port 10-Gigabit Ethernet IQ2 and IQ2-EPICs</td>
<td>Interface</td>
<td>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.</td>
<td>The Layer 2 overhead is 18 bytes for non-VLAN traffic and 22 bytes for VLAN traffic.</td>
</tr>
<tr>
<td>4-port 1G IQ2 and IQ2-E PICs Packet forwarding component</td>
<td>Queued: 478 bytes of Layer 3 packet. Transmitted: 478 bytes of Layer 3 packet.</td>
<td>–</td>
<td></td>
</tr>
<tr>
<td>8-port 1G IQ2 and IQ2-E PICs</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Physical interface: ge-4/2/0, Enabled, Physical link is Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface index: 203, SNMP ifIndex: 1054</td>
</tr>
<tr>
<td>Forwarding classes: 16 supported, 4 in use</td>
</tr>
<tr>
<td>Egress queues: 8 supported, 4 in use</td>
</tr>
<tr>
<td>Queue: 0, Forwarding classes: best-effort</td>
</tr>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 131300649, 141751 pps</td>
</tr>
<tr>
<td>Bytes : 11287964840, 99793248 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 131300649, 141751 pps</td>
</tr>
<tr>
<td>Bytes : 11287964840, 99793248 bps</td>
</tr>
<tr>
<td>Tail-dropped packets : 0, 0 pps</td>
</tr>
<tr>
<td>RL-dropped packets : 205050862, 602295 pps</td>
</tr>
<tr>
<td>RL-dropped bytes : 13595326612, 327648832 bps</td>
</tr>
<tr>
<td>RED-dropped packets : 0, 0 pps</td>
</tr>
<tr>
<td>Low : 0, 0 pps</td>
</tr>
<tr>
<td>Medium-low : 0, 0 pps</td>
</tr>
<tr>
<td>Medium-high : 0, 0 pps</td>
</tr>
<tr>
<td>High : 0, 0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes : 0, 0 bps</td>
</tr>
<tr>
<td>Low : 0, 0 bps</td>
</tr>
<tr>
<td>Medium-low : 0, 0 bps</td>
</tr>
<tr>
<td>Medium-high : 0, 0 bps</td>
</tr>
<tr>
<td>High : 0, 0 bps</td>
</tr>
<tr>
<td>Queue: 1, Forwarding classes: expedited-forwarding</td>
</tr>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 0, 0 pps</td>
</tr>
<tr>
<td>Bytes : 0, 0 bps</td>
</tr>
</tbody>
</table>

**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
```

<table>
<thead>
<tr>
<th>Physical interface: ae1, Enabled, Physical link is Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface index: 158, SNMP ifIndex: 33</td>
</tr>
<tr>
<td>Forwarding classes: 8 supported, 8 in use</td>
</tr>
<tr>
<td>Output queues: 8 supported, 8 in use</td>
</tr>
<tr>
<td>Queue: 0, Forwarding classes: be</td>
</tr>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 5, 0 pps</td>
</tr>
<tr>
<td>Bytes : 242, 0 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 5, 0 pps</td>
</tr>
<tr>
<td>Bytes : 242, 0 bps</td>
</tr>
<tr>
<td>Tail-dropped packets : 0, 0 pps</td>
</tr>
<tr>
<td>RED-dropped packets : 0, 0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes : 0, 0 bps</td>
</tr>
<tr>
<td>Queue: 1, Forwarding classes: af1</td>
</tr>
<tr>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 42603765</td>
</tr>
<tr>
<td>Bytes : 5453281920</td>
</tr>
<tr>
<td>Packets : 42603765</td>
</tr>
<tr>
<td>Bytes : 5453281920</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 42603765</td>
</tr>
<tr>
<td>Bytes : 5453281920</td>
</tr>
<tr>
<td>Tail-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped bytes : 0</td>
</tr>
</tbody>
</table>
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      : 22156199818        204742296 bps
  Transmitted:
    Packets    : 4850003             4033 pps
    Bytes      : 2619000162          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:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4836136</td>
<td>33340232</td>
<td>3912 pps</td>
</tr>
</tbody>
</table>

### Transmitted:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3600866</td>
<td>24485888</td>
<td>1459 pps</td>
</tr>
</tbody>
</table>

### Tail-dropped packets: Not Available

### RED-dropped packets: 1225034 (2450 pps)

### RED-dropped bytes: 83302312 (1333072 bps)

---

### Queue: 3, Forwarding classes: network-control

#### Queued:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0 pps</td>
</tr>
</tbody>
</table>

#### Transmitted:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0 pps</td>
</tr>
</tbody>
</table>

---

### Packet Forwarding Engine Chassis Queues:

#### Queues: 4 supported, 4 in use

#### Queue: 0, Forwarding classes: best-effort

#### Queued:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>77059796</td>
<td>3544750624</td>
<td>486384 pps</td>
</tr>
</tbody>
</table>

#### Transmitted:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>77059797</td>
<td>3544750670</td>
<td>486381 pps</td>
</tr>
</tbody>
</table>

---

#### Queue: 1, Forwarding classes: expedited-forwarding

#### Queued:

<table>
<thead>
<tr>
<th>Packets</th>
<th>Bytes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0 pps</td>
</tr>
</tbody>
</table>
Transmitted:
<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bytes</td>
<td></td>
<td>BPS</td>
<td></td>
</tr>
<tr>
<td>Tail-dropped</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>RED-dropped</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
</tbody>
</table>

Queue: 2, Forwarding classes: assured-forwarding

Queued:
<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bytes</td>
<td></td>
<td>BPS</td>
<td></td>
</tr>
<tr>
<td>Tail-dropped</td>
<td>4846580</td>
<td>222942680</td>
<td>3934 pps</td>
<td>1447768 bps</td>
</tr>
<tr>
<td>RED-dropped</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
</tbody>
</table>

Queue: 3, Forwarding classes: network-control

Queued:
<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bytes</td>
<td></td>
<td>BPS</td>
<td></td>
</tr>
<tr>
<td>Tail-dropped</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>RED-dropped</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
<td>0 pps</td>
<td></td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
<td>0 bps</td>
<td></td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Queue: 1, Forwarding classes: expedited-forwarding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 4,183,900,390  10 pps</td>
</tr>
<tr>
<td>Bytes : 389,102,697,520  7,440 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 4,183,900,390  10 pps</td>
</tr>
<tr>
<td>Bytes : 389,102,697,520  7,440 bps</td>
</tr>
<tr>
<td>Tail-dropped packets: Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0  0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes : 0  0 bps</td>
</tr>
<tr>
<td>Queue: 2, Forwarding classes: assured-forwarding</td>
</tr>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 0  0 pps</td>
</tr>
<tr>
<td>Bytes : 0  0 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0  0 pps</td>
</tr>
<tr>
<td>Bytes : 0  0 bps</td>
</tr>
<tr>
<td>Tail-dropped packets: Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0  0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes : 0  0 bps</td>
</tr>
<tr>
<td>Queue: 3, Forwarding classes: network-control</td>
</tr>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : 7,055  1 pps</td>
</tr>
<tr>
<td>Bytes : 451,552  512 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 7,055  1 pps</td>
</tr>
<tr>
<td>Bytes : 451,552  512 bps</td>
</tr>
<tr>
<td>Tail-dropped packets: Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0  0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes : 0  0 bps</td>
</tr>
</tbody>
</table>

Forwarding classes: 16 supported, 4 in use Egress queues: 4 supported, 4 in use

Queue: 0, Forwarding classes: best-effort

Queued:
| Packets : 1,031  0 pps |
| Bytes : 143,292  0 bps |
| Transmitted:                                  |
| Packets : 1,031  0 pps |
| Bytes : 143,292  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           |
### 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

### 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

### Packet Forwarding Engine Chassis Queues:

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: Not Available

### Queue: 1, Forwarding classes: expedited-forwarding

**Queued:**
- Packets: 0, 0 pps
- Bytes: 0, 0 bps

**Transmitted:**
- Packets: 0, 0 pps
- Bytes: 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 bps

  Transmitted:
  Packets : 254 0 pps
  Bytes : 16274 0 bps
<table>
<thead>
<tr>
<th>Queue: 1, Forwarding classes: expedited-forwarding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : Not Available</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Tail-dropped packets : Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped bytes : 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue: 2, Forwarding classes: assured-forwarding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : Not Available</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Tail-dropped packets : Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped bytes : 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue: 3, Forwarding classes: network-control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : Not Available</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Tail-dropped packets : Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped bytes : 0</td>
</tr>
</tbody>
</table>

Forwarding classes: 8 supported, 4 in use
Egress queues: 4 supported, 4 in use

<table>
<thead>
<tr>
<th>Queue: 0, Forwarding classes: best-effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : Not Available</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 3</td>
</tr>
<tr>
<td>Bytes  : 126</td>
</tr>
<tr>
<td>Tail-dropped packets : Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped bytes : 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue: 1, Forwarding classes: expedited-forwarding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : Not Available</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Tail-dropped packets : Not Available</td>
</tr>
<tr>
<td>RED-dropped packets : 0</td>
</tr>
<tr>
<td>RED-dropped bytes : 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queue: 2, Forwarding classes: assured-forwarding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued:</td>
</tr>
<tr>
<td>Packets : Not Available</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes  : 0</td>
</tr>
<tr>
<td>Queue</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Layer 2 Bridging, Address Learning, and Forwarding Feature Guide
### show interfaces queue ingress (Gigabit Ethernet IQ2 PIC)

```plaintext
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)

```plaintext
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
```
<table>
<thead>
<tr>
<th>Queue</th>
<th>Forwarding classes</th>
<th>Queued:</th>
<th>Transmitted:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Packets: Not Available</td>
<td>Packets: 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bytes: 0</td>
<td>Bytes: 126</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tail-dropped packets: Not Available</td>
<td>Tail-dropped packets: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RED-dropped packets: 0</td>
<td>RED-dropped packets: 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Queue: 1, Forwarding classes: expedited-forwarding</td>
<td>Queue: 1, Forwarding classes: assured-forwarding</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Queue: 2, Forwarding classes: network-control</td>
<td>Queue: 3, Forwarding classes: best-effort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Queue: 0, Forwarding classes: expedited-forwarding</td>
<td>Queue: 0, Forwarding classes: best-effort</td>
</tr>
</tbody>
</table>

**Packet Forwarding Engine Chassis Queues:**

- **Queue: 0,** Forwarding classes: best-effort
  - Queued:
    - Packets: 80564692
    - Bytes: 3383717100
  - Transmitted:
    - Packets: 80564692
    - Bytes: 3383717100

- **Queue: 1,** Forwarding classes: expedited-forwarding
  - Queued:
    - Packets: 80564685
    - Bytes: 3383716770
  - Transmitted:
    - Packets: 80564685
    - Bytes: 3383716770

- **Queue: 2,** Forwarding classes: assured-forwarding
  - Queued:
    - Packets: Not Available
    - Bytes: 0
  - Transmitted:
    - Packets: Not Available
    - Bytes: 0

- **Queue: 3,** Forwarding classes: network-control
  - Queued:
    - Packets: Not Available
    - Bytes: 0
  - Transmitted:
    - Packets: Not Available
    - Bytes: 0

- **Queue: 4** supported, 4 in use

---

**Layer 2 Bridging, Address Learning, and Forwarding Feature Guide**
<table>
<thead>
<tr>
<th></th>
<th>Queue: 2, Forwarding classes: assured-forwarding</th>
<th>Queue: 3, Forwarding classes: network-control</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED-dropped bytes</td>
<td>0 bps</td>
<td>0 bps</td>
</tr>
<tr>
<td>Queued:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>9538</td>
</tr>
<tr>
<td>Bytes</td>
<td>0 bps</td>
<td>3819840</td>
</tr>
<tr>
<td>Transmitted:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>9538</td>
</tr>
<tr>
<td>Bytes</td>
<td>0 bps</td>
<td>3819840</td>
</tr>
</tbody>
</table>

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 bps
Transmitted:
  Packets : 0 0 pps
  Bytes : 0 bps
Tail-dropped packets : Not Available
RED-dropped packets : 0 0 pps
  Low : 0 0 pps  
  Medium-low : 0 0 pps  
RED-dropped bytes : 0 0 bps
  Low : 0 0 bps  
  Medium-low : 0 0 bps
```
| 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

<table>
<thead>
<tr>
<th>Queued:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 bps</td>
</tr>
</tbody>
</table>

Queue: 3, Forwarding classes: network-control

<table>
<thead>
<tr>
<th>Queued:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 bps</td>
</tr>
</tbody>
</table>

Forwarding classes: 16 supported, 4 in use
Egress queues: 4 supported, 4 in use

Queue: 0, Forwarding classes: best-effort

<table>
<thead>
<tr>
<th>Queued:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>109355853</td>
<td>471736 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>7436199152</td>
<td>256627968 bps</td>
</tr>
<tr>
<td>Transmitted:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>109355852</td>
<td>471736 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>7436198640</td>
<td>256627968 bps</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 bps</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>Category</th>
<th>Queued</th>
<th>Transmitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Queue: 2, Forwarding classes: PRIVATE

Command Output:

Queue: 2, Forwarding classes: PRIVATE

<table>
<thead>
<tr>
<th>Category</th>
<th>Queue: 2</th>
<th>Transmitted:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Queue</td>
<td>Forwarding classes:</td>
<td>Packets</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------</td>
<td>---------</td>
</tr>
<tr>
<td>3</td>
<td>CONTROL</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>CLASS_B_OUTPUT</td>
<td>0</td>
</tr>
<tr>
<td>Class</td>
<td>Queued Packets</td>
<td>Queued Bytes</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Queue: 5, Forwarding classes: CLASS_C_OUTPUT

Queued:
- Packets: 0 pps
- Bytes: 0 bps

Transmitted:
- Packets: 0 pps
- Bytes: 0 bps

Queue: 6, Forwarding classes: CLASS_V_OUTPUT

Queued:
- Packets: 0 pps
- Bytes: 0 bps

Transmitted:
<table>
<thead>
<tr>
<th>Category</th>
<th>Queued:</th>
<th>Transmitted:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Queue: 7, Forwarding classes:</td>
<td>CLASS_S_OUTPUT, GETS</td>
<td></td>
</tr>
</tbody>
</table>

Queue: 7, Forwarding classes: CLASS_S_OUTPUT, GETS
<table>
<thead>
<tr>
<th></th>
<th>Queued</th>
<th>Transmitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Packets: 371365</td>
<td>Packets: 371365</td>
</tr>
<tr>
<td></td>
<td>Bytes: 15597330</td>
<td>Bytes: 15597330</td>
</tr>
<tr>
<td></td>
<td>Tail-dropped packets: 0</td>
<td>Tail-dropped packets: 0</td>
</tr>
<tr>
<td></td>
<td>RED-dropped packets: 0</td>
<td>RED-dropped packets: 0</td>
</tr>
<tr>
<td></td>
<td>Low: 0</td>
<td>Low: 0</td>
</tr>
<tr>
<td></td>
<td>Medium-low: 0</td>
<td>Medium-low: 0</td>
</tr>
<tr>
<td></td>
<td>Medium-high: 0</td>
<td>Medium-high: 0</td>
</tr>
<tr>
<td></td>
<td>High: 0</td>
<td>High: 0</td>
</tr>
<tr>
<td></td>
<td>RED-dropped bytes: 0</td>
<td>RED-dropped bytes: 0</td>
</tr>
<tr>
<td></td>
<td>Low: 0</td>
<td>Low: 0</td>
</tr>
<tr>
<td></td>
<td>Medium-low: 0</td>
<td>Medium-low: 0</td>
</tr>
<tr>
<td></td>
<td>Medium-high: 0</td>
<td>Medium-high: 0</td>
</tr>
<tr>
<td></td>
<td>High: 0</td>
<td>High: 0</td>
</tr>
</tbody>
</table>

Packet Forwarding Engine Chassis Queues:

Queues: 8 supported, 8 in use

Queue 0, Forwarding classes: DEFAULT

Queued:

|        | Packets: 371365 | Packets: 371365 |
|        | Bytes: 15597330 | Bytes: 15597330 |
|        | Tail-dropped packets: 0 | Tail-dropped packets: 0 |
|        | RED-dropped packets: 0 | RED-dropped packets: 0 |
|        | Low: 0 | Low: 0 |
|        | Medium-low: 0 | Medium-low: 0 |
|        | Medium-high: 0 | Medium-high: 0 |
|        | High: 0 | High: 0 |
|        | RED-dropped bytes: 0 | RED-dropped bytes: 0 |
|        | Low: 0 | Low: 0 |
|        | Medium-low: 0 | Medium-low: 0 |
|        | Medium-high: 0 | Medium-high: 0 |
|        | High: 0 | High: 0 |

Queue 1, Forwarding classes: REALTIME

Queued:

<p>|        | Packets: 0 | Packets: 0 |
|        | Bytes: 0 | Bytes: 0 |
|        | Tail-dropped packets: 0 | Tail-dropped packets: 0 |</p>
<table>
<thead>
<tr>
<th></th>
<th>Queue: 2, Forwarding classes: PRIVATE</th>
<th>Queue: 3, Forwarding classes: CONTROL</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Queued:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Transmitted:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Chapter 11: Operational Mode Commands for Layer 2 Bridge Domains

Queue: 4, Forwarding classes: CLASS_B_OUTPUT

Queued:

<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Transmitted:

<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Transmitted:

<table>
<thead>
<tr>
<th></th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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
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
<table>
<thead>
<tr>
<th>Category</th>
<th>Packets</th>
<th>PPS</th>
<th>Bytes</th>
<th>BPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0 pps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 pps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 pps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 pps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 pps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 bps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 bps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 bps</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 bps</td>
<td>0</td>
<td>0 bps</td>
</tr>
</tbody>
</table>

Queue: 7, Forwarding classes: CLASS_S_OUTPUT, GETS

Queued:

- Packets: 0 pps
- Bytes: 0 bps

Transmitted:

- Packets: 0 pps
- Bytes: 0 bps
- Tail-dropped packets: 0 pps
- RED-dropped packets: 0 pps
- Low: 0 pps
- Medium-low: 0 pps
- Medium-high: 0 pps
- High: 0 pps
- RED-dropped bytes: 0 bps
- Low: 0 bps
- Medium-low: 0 bps
- Medium-high: 0 bps
- High: 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
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

<table>
<thead>
<tr>
<th>Queued:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets : 55576</td>
</tr>
<tr>
<td>Bytes : 29622008</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 55576</td>
</tr>
<tr>
<td>Bytes : 29622008</td>
</tr>
</tbody>
</table>

Tail-dropped packets : 0 pps
RL-dropped packets : 0 pps
RL-dropped bytes : 0 bps
RED-dropped packets : 0 pps
Low : 0 pps
Medium-low : 0 pps
Medium-high : 0 pps
High : 0 pps
RED-dropped bytes : 0 bps
Low : 0 bps
Medium-low : 0 bps
Medium-high : 0 bps
High : 0 bps

Queue: 1, Forwarding classes: ef

<table>
<thead>
<tr>
<th>Queued:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes : 0</td>
</tr>
</tbody>
</table>

Tail-dropped packets : 0 pps
RL-dropped packets : 0 pps
RL-dropped bytes : 0 bps
RED-dropped packets : 0 pps
Low : 0 pps
Medium-low : 0 pps
Medium-high : 0 pps
High : 0 pps
RED-dropped bytes : 0 bps
Low : 0 bps
Medium-low : 0 bps
Medium-high : 0 bps
High : 0 bps

Queue: 2, Forwarding classes: af

<table>
<thead>
<tr>
<th>Queued:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes : 0</td>
</tr>
<tr>
<td>Transmitted:</td>
</tr>
<tr>
<td>Packets : 0</td>
</tr>
<tr>
<td>Bytes : 0</td>
</tr>
</tbody>
</table>

Tail-dropped packets : 0 pps
RL-dropped packets : 0 pps
RL-dropped bytes : 0 bps
RED-dropped packets : 0 pps
Low : 0 pps
Medium-low : 0 pps
Medium-high : 0 pps
High : 0 pps
RED-dropped bytes : 0 bps
Low : 0 bps
Medium-low : 0 bps
Medium-high : 0 bps
High : 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 : 11901376

Queue: 1, Forwarding classes: expedited-forwarding

Queued:
  Packets : 0  0 pps
<table>
<thead>
<tr>
<th>Bytes</th>
<th>0</th>
<th>0 bps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Tail-dropped packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RL-dropped packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RL-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>RED-dropped packets</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Queue-depth bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Peak</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>32768</td>
<td></td>
</tr>
</tbody>
</table>

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

<p>| 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 bps |
| High | 0 | 0 bps |</p>
<table>
<thead>
<tr>
<th>Category</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 pps</td>
</tr>
<tr>
<td>RED-dropped bytes</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-low</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Medium-high</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>High</td>
<td>0</td>
<td>0 bps</td>
</tr>
<tr>
<td>Queue-depth bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Current</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Peak</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>6258688</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 12

Operational Mode Commands for Layer 2 Learning

- clear l2-learning mac-move-buffer
- show l2-learning global-information
- show l2-learning global-mac-count
- show l2-learning instance
- show l2-learning interface
- show l2-learning mac-move-buffer
### clear l2-learning mac-move-buffer

**Syntax**

```
clear l2-learning mac-move-buffer
<active>
```

**Release Information**

Command introduced in Junos OS Release 13.2.

**Description**

Clear the MAC move buffer entries.

**Options**

- **none**— Clear the MAC move buffer entries.
- **active**— (Optional) Unblock the interfaces that were blocked by the 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, the user can manually release the blocking.

**Required Privilege**

`clear`

**List of Sample Output**

- `clear l2-learning mac-move-buffer` on page 224
- `clear l2-learning mac-move-buffer active` on page 224

**Output Fields**

When you enter this command, the MAC move buffer entries are deleted.

**Sample Output**

```
clear l2-learning mac-move-buffer
user@host> clear l2-learning mac-move-buffer

clear l2-learning mac-move-buffer active
user@host> clear l2-learning mac-move-buffer active
```
**show l2-learning global-information**

**Syntax**

```
show l2-learning global-information
```

**Release Information**

Command introduced in Junos OS Release 8.4.

**Description**

(MX Series routers only) Display Layer 2 learning process-related information for the entire router.

**Options**

This command has no options.

**Required Privilege Level**

view

**List of Sample Output**

show l2-learning global-information on page 225

**Output Fields**

Table 11 on page 225 describes the output fields for the `show l2-learning global-information` command. Output fields are listed in the approximate order in which they appear.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC aging interval</td>
<td>Configured timeout interval, in seconds, for all MAC table entries.</td>
</tr>
<tr>
<td>MAC learning</td>
<td>Status of MAC learning: Enabled or Disabled.</td>
</tr>
<tr>
<td>MAC statistics</td>
<td>Status of MAC accounting: Enabled or Disabled.</td>
</tr>
<tr>
<td>MAC limit Count</td>
<td>Configured maximum limit on the number of MAC addresses that can be learned.</td>
</tr>
<tr>
<td>MAC limit hit flag</td>
<td>Status of the learned MAC limit hit flag: Enabled (the learned MAC exceeds the global MAC limit) or Disabled (the learned MAC does not exceed the global MAC limit).</td>
</tr>
<tr>
<td>MAC packet action drop</td>
<td>Status of action to drop packets after the configured MAC address limit is reached: Enabled (packets are dropped) or Disabled (packets are forwarded).</td>
</tr>
</tbody>
</table>

**Sample Output**

```
show l2-learning global-information

user@host> show l2-learning global-information

Global Configuration:

MAC aging interval : 300
```
<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC learning</td>
<td>Enabled</td>
</tr>
<tr>
<td>MAC statistics</td>
<td>Disabled</td>
</tr>
<tr>
<td>MAC limit Count</td>
<td>393215</td>
</tr>
<tr>
<td>MAC limit hit flag</td>
<td>Disabled</td>
</tr>
<tr>
<td>MAC packet action drop</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
### show l2-learning global-mac-count

<table>
<thead>
<tr>
<th>Syntax</th>
<th>show l2-learning global-mac-count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Information</td>
<td>Command introduced in Junos OS Release 9.3.</td>
</tr>
<tr>
<td>Description</td>
<td>(MX Series routers only) Display the total number of dynamic and static MAC addresses learned for the entire router.</td>
</tr>
<tr>
<td>Options</td>
<td>This command has no options.</td>
</tr>
<tr>
<td>Required Privilege Level</td>
<td>view</td>
</tr>
</tbody>
</table>

**List of Sample Output**

show l2-learning global-mac-count on page 227

**Output Fields**

Displays the total number of dynamic and static MAC addresses learned for the entire router.

### Sample Output

```bash
user@host> show l2-learning global-mac-count
100 dynamic and static MAC addresses learned globally
```
show l2-learning instance

Syntax
show l2-learning instance

Release Information
(MX Series routers only) Command introduced in Junos OS Release 8.4.

Description
Display Layer 2 learning properties for all the configured routing instances.

Options
This command has no options.

Required Privilege Level
view

List of Sample Output
show l2-learning instance on page 229

Output Fields
Table 12 on page 228 describes the output fields for the show l2-learning instance command. Output fields are listed in the approximate order in which they appear.

Table 12: show l2-learning instance Output Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Instance</td>
<td>Name of routing instance.</td>
</tr>
<tr>
<td>Bridging Domain</td>
<td>Name of bridging domain.</td>
</tr>
<tr>
<td></td>
<td>On MX Series routers you can use the show l2-learning instance &lt;extensive&gt; command option to display the Bridge Service-id information which includes the Config Service ID and the Active Service ID.</td>
</tr>
<tr>
<td>Index</td>
<td>Number associated with the routing instance or bridging domain.</td>
</tr>
<tr>
<td>Logical System</td>
<td>Name of logical system or Default if no logical system is configured.</td>
</tr>
<tr>
<td>Routing instance flags</td>
<td>Status of Layer 2 learning properties for each routing instance:</td>
</tr>
<tr>
<td></td>
<td>• DL—MAC learning is disabled.</td>
</tr>
<tr>
<td></td>
<td>• SE—MAC accounting is enabled.</td>
</tr>
<tr>
<td></td>
<td>• AD—Packets are dropped after MAC address limit is reached.</td>
</tr>
<tr>
<td></td>
<td>• LH—the maximum number of MAC addresses has been learned on the routing instance. The routing instance is not able to learn any additional MAC addresses.</td>
</tr>
<tr>
<td>MAC limit</td>
<td>Maximum number of MAC addresses that can be learned from each interface in the routing instance or bridging domain.</td>
</tr>
</tbody>
</table>
Sample Output

show l2-learning instance

user@host> show l2-learning instance

Information for routing instance:

Routing Instance flags (DL -disable learning, SE -stats enabled,
AD -packet action drop, LH -mac limit hit)

<table>
<thead>
<tr>
<th>Routing Instance</th>
<th>Bridging</th>
<th>Index</th>
<th>Logical</th>
<th>Routing</th>
<th>MAC limit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>juniper_private1</strong></td>
<td></td>
<td>1</td>
<td>Default</td>
<td></td>
<td>5000</td>
</tr>
<tr>
<td>vs1</td>
<td>vlan100</td>
<td>3</td>
<td>Default</td>
<td></td>
<td>5120</td>
</tr>
<tr>
<td>vs1</td>
<td>vlan200</td>
<td>4</td>
<td>Default</td>
<td></td>
<td>5120</td>
</tr>
</tbody>
</table>
**show l2-learning interface**

**Syntax**

```
show l2-learning interface
```

**Release Information**

Command introduced in Junos OS Release 8.4. Added sample output to indicate an EVPN MAC Pinned interface, introduced in Junos OS 16.2R1.

**Description**

(MX Series routers only) Display Layer 2 learning information for all the interfaces.

**Options**

This command has no options.

**Required Privilege Level**

view

**List of Sample Output**

show l2-learning interface on page 230
show l2-learning-interface on page 231

**Output Fields**

Table 13 on page 230 describes the output fields for the `show l2-learning interface` command. Output fields are listed in the approximate order in which they appear.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical interface</td>
<td>Name of the logical interface.</td>
</tr>
<tr>
<td>Index</td>
<td>Index of the interface.</td>
</tr>
<tr>
<td>Routing Instance</td>
<td>Number of the routing instance to which the interface belongs.</td>
</tr>
<tr>
<td>Interface device</td>
<td>Value of the order in which the Junos OS finds and initializes the interface.</td>
</tr>
<tr>
<td>Logical interface flags</td>
<td>Status of Layer 2 learning properties for each interface:</td>
</tr>
<tr>
<td></td>
<td>• DL—MAC learning is disabled.</td>
</tr>
<tr>
<td></td>
<td>• SE—MAC accounting is enabled.</td>
</tr>
<tr>
<td></td>
<td>• AD—Packets are dropped after the MAC interface limit is reached.</td>
</tr>
<tr>
<td></td>
<td>• MAC limit—Maximum number of MAC addresses that can be learned from the interface.</td>
</tr>
<tr>
<td></td>
<td>• MP—MAC Pinning enabled.</td>
</tr>
</tbody>
</table>

**Sample Output**

```
show l2-learning interface
```

```
user@host> show l2-learning interface
```
Information for interface family:

Logical Interface flags (DL -disable learning, SE -stats enabled, AD -packet action drop, LH -mac limit hit)

<table>
<thead>
<tr>
<th>Logical interface</th>
<th>Index</th>
<th>Routing instance</th>
<th>Interface device</th>
<th>Logical Interface flags</th>
<th>MAC limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ge-11/0/3.0</td>
<td>79</td>
<td>3</td>
<td>136</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>ge-11/1/4.100</td>
<td>84</td>
<td>3</td>
<td>150</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>ge-11/1/1.100</td>
<td>86</td>
<td>3</td>
<td>147</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>ge-11/1/0.100</td>
<td>87</td>
<td>3</td>
<td>146</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>xe-10/2/0.100</td>
<td>88</td>
<td>3</td>
<td>144</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>xe-10/0/0.100</td>
<td>89</td>
<td>3</td>
<td>129</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>ge-11/1/0.200</td>
<td>90</td>
<td>4</td>
<td>146</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>ge-11/1/1.200</td>
<td>91</td>
<td>4</td>
<td>147</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>ge-11/1/4.200</td>
<td>92</td>
<td>4</td>
<td>150</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>xe-10/0/0.200</td>
<td>93</td>
<td>4</td>
<td>129</td>
<td></td>
<td>1024</td>
</tr>
<tr>
<td>xe-10/2/0.200</td>
<td>94</td>
<td>4</td>
<td>144</td>
<td></td>
<td>1024</td>
</tr>
</tbody>
</table>

show l2 learning-interface

user@host> run show l2-learning interface

Routing Instance Name: default-switch
Logical Interface flags (DL -disable learning, AD -packet action drop, LH - MAC limit hit, DN - Interface Down, MP - MAC Pinning enabled)

<table>
<thead>
<tr>
<th>Logical Interface</th>
<th>BD</th>
<th>MAC</th>
<th>STP</th>
<th>Logical Interface flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>ae0.0</td>
<td></td>
<td>8192</td>
<td></td>
<td>MP</td>
</tr>
</tbody>
</table>

Copyright © 2018, Juniper Networks, Inc.
# show l2-learning mac-move-buffer

## Syntax
```
show l2-learning mac-move-buffer
  <brief | detail | extensive>
  <active>
```

## Release Information
Command introduced in Junos OS Release 9.4.

## Description
(MX Series routers only) Display action as a result of configuring the MAC address move feature.

## Options
- `none`— Display action as a result of the MAC address move feature.
- `brief | detail | extensive`— (Optional) Display the specified level of output.
- `active`— (Optional) Display the set of interfaces blocked as a result of the MAC address move action.

## Required Privilege
- **Level** view

## List of Sample Output
- `show l2-learning mac-move-buffer active on page 232`
- `show l2-learning mac-move-buffer extensive on page 232`

## Output Fields
Display action as a result of the MAC address move feature.

## Sample Output
### show l2-learning mac-move-buffer active
```
user@host> show l2-learning mac-move-buffer active

MAC Address: 00:00:00:00:01:01, VLAN Id: 0
  Time Rec: 2012-06-25 06:23:41   Bridge Domain: bd10
  Prev IFL : ge-1/0/5.0            New IFL: ge-1/0/6.0
  IFBD     : ge-1/0/6.0:10         Blocked : YES
```

### show l2-learning mac-move-buffer extensive
```
user@host> show l2-learning mac-move-buffer extensive | display xml

<12ald-mac-move-buffer>
  <12ald-mac-move-entry junos:style="extensive">
    <12ald-mac-address>aa:00:00:00:02:00</12ald-mac-address>
    <12ald-learn-vlan-id>0</12ald-learn-vlan-id>
    <12ald-mac-move-bridge-domain>bd</12ald-mac-move-bridge-domain>
    <12ald-mac-move-from-ifl>ge-1/0/5.200</12ald-mac-move-from-ifl>
    <12ald-mac-move-to-ifl>ge-1/0/6.200</12ald-mac-move-to-ifl>
    <12ald-mac-move-to-ifbd>ge-1/0/6.200</12ald-mac-move-to-ifbd>
  </12ald-mac-move-entry>
</12ald-mac-move-buffer>
```
<l2ald-mac-move-is-blocked>Yes</l2ald-mac-move-is-blocked>
</l2ald-mac-move-entry>